SPL Implementation Guide for FDA Content of Labeling Submissions

Version 2a
March, 2005

Interim FDA Document

Principal Contributors:

Lori Baranowksi, Bristol Myers Squibb
Sandy Boyer, Boyer-Boyer Inc.
Pamela Budny, Eli Lilly Co.
Glenda Casper, Wyeth, Inc.
Steven Gitterman, US FDA (Principal Editor)
Yoshi Murata, US FDA
Toni Stifano, US FDA
Gunther Schadow, Indiana University
Keith Thomas, Infastructures for Information
Robert Wallace, Eli Lilly Co.

Questions or comments regarding this document should be directed to Steven Gitterman at *steven.gitterman@fda.gov*

Note: This document is an FDA-authored interim revision of the HL7 document, "SPL Implementation Guide for FDA Content of Labeling Submissions, Release 1". This document reflects the differences between SPL Release 1 and the SPL schema ('2a') that has been released by FDA for validating SPL submissions to FDA (see http://www.fda.gov/oc/datacouncil/spl.html). The principal contributors listed above contributed to the original Release 1 document, not the modifications included in this document.

This document is NOT an HL7 informative document. All comments regarding this document should be directed to steven.gitterman@fda.gov and not HL7. It is expected that after the balloting process for SPL Release 2 is completed this document will be revised and submitted for reballoting to HL7. It is also likely that this document will be revised often over the next few months in response to comments regading this document and as more information becomes available about code systems referred to in this document but not finalized at this time. Because of the interim status of this document and the expectation it will be superceded by an HL7 document, it does not conform to the usual style convention of FDA publications.

FDA will be publishing a separate reference document addressing versioning and life cycle requirements for SPL. It is anticipated this will be available in early May at http://www.fda.gov/oc/datacouncil/spl.html.

The authors wish to acknowledge the substantial help of Binh Ta in the editing of this revision.

Table of Contents

List of Figures:

List of Tables:

1.

Introduction

The Structured Product Labeling Implementation Guide for FDA Content of Labeling Submissions (SPL Implementation Guide) is a companion to the Health Level Seven (HL7) Structured Product Labeling (SPL) normative standard. HL7 is one of several American National Standards Institute (ANSI) accredited Standards Developing Organizations (SDO) for health care. The latest approved version of SPL Schema (Release 1), which is the strict technical definition of an SPL document, is available from HL7 at http://www.hl7.org/Special/committees/rcrim/docs.cfm.[1]

The HL7 Informative Document, SPL Implementation Guide for FDA Content of Labeling Submissions Release 1, was originally created by the HL7 SPL working group specifically to provide additional information for creating Content of Labeling submissions to the FDA described in the guidance to industry: *Guidance for Industry - Providing Regulatory Submissions in Electronic Format – Content of Labeling*[2].

At the present time (March, 2005), SPL Release 2 is undergoing membership balloting at HL7. SPL Release 2 contains a number of changes from Release 1 and is not fully backward compatible with Release 1. FDA has released a schema labeled 'SPL 2a' which is a subset of the full SPL Release 2 balloted schema; this schema includes changes to Release 1 that FDA believes are important to implementation of SPL. No objections or comments to the changes included in SPL 2a were raised during committee level balloting of the full SPL Release 2, and FDA expects the changes reflected in the 'FDA SPL schema' will be adopted unchanged as part of the full Release 2 schema.

This document is a revision of the HL7 SPL Implementation Guide for FDA Content of Labeling Submissions Release 1 to address the changes in the FDA SPL schema. This document is not an HL7 informative document although differing only slightly from the HL7 SPL Implementation Guide Release 1.

2. Creating an SPL Document[3]

2.1 Introduction

Structured Product Labeling (SPL) is the HL7 standard for describing the content of prescription drug labeling in an XML document. An SPL document consists of an XML (extensible markup language) document that contains the text and images in an approved prescription package insert (i.e., the content of labeling) along with additional information for machine processing of label content (i.e., header information and data elements, described below). The SPL XML file is converted to a human-readable format by the use of a set of files collectively referred to as a stylesheet. The stylesheet displays the information in the XML file in a consistent format for viewing. Currently, the standard stylesheet supports display in web browsers only.

An SPL document may be created using a variety of possible editing environments, ranging from a general purpose word processor to an XML editor to a SPL-specific editing tool. Although considerable differences in the approach to creating an SPL document are determined by the choice of environment, the final document will be independent of the tool used for creation; all will be expected to be valid against the SPL schema as defined by the SPL standard.[4] It is also anticipated that where options exist in creating the actual SPL document, 'Best Practices' and regulatory requirements will be followed.

This section specifically addresses creating an SPL document outside of a dedicated SPL-specific environment, i.e., using a text-processing environment or a general XML-oriented editing environment. Ideally, use of a dedicated 'SPL creation' tool will 'blind' the SPL author to many of the details addressed in

[1] A brief overview of XML that includes a definition of XML schemas is included in Appendix 7.5.
[2] See Guidance for Industry: Providing Regulatory Submissions in Electronic Format — Content of Labeling (February, 2004).
[3] This guide follows the standards for SPL proposed in the December, 2004 HL7 ballot for Structured Product Labeling Release 2 (available at http://www.hl7.org). Changes from SPL Release 1 will be discussed when relevant.
[4] Validity of an XML document is discussed in the XML Primer in Appendix 7.5

this section. This section may also be of interest to developers or individuals engaged in the quality control of SPL documents.

The HL7 SPL stylesheet is the method adopted by HL7 to produce a standard display[5] of SPL, i.e., a common 'appearance' of all SPL documents when displayed. At present, the stylesheet is specific for viewing SPL in a web browser.[6] SPL also provides a mechanism whereby additional display 'styles' can be added when SPL is used in different regulatory environments. For US regulatory purposes, FDA will maintain (as necessary) a separate stylesheet which 'adds' to the HL7 standard stylesheet for the display of the content of labeling for documents submitted to FDA. Specific details regarding the FDA-developed stylesheet are described in Appendix 7.2: SPL Standard Stylesheet and FDA Implementation. For conceptual purposes, the display of SPL in a web browser can be considered simply as drawing from 'styles' that are available in both the HL7 and FDA stylesheets; whenever reference is made to display by the 'standard stylesheet', in effect this refers to the HL7 standard stylesheet.

This SPL rendition (i.e., display by the 'standard stylesheet') presents only the content of labeling contained in SPL for viewing by the user. Additional information in SPL, e.g., header information (see Sec. 3.2) and data elements are not part of this display. Other stylesheets are likely to be available which will highlight these features based on individual needs.

SPL has been developed as a document format to transmit the content of labeling rather than a mechanism for reproducing the exact format of printed package inserts. The standard stylesheet specifies the default font, indentation, orientation, formatting, word wrapping, line spacing, and other properties that will be used for the 'standard' display. Formatting (cascading stylesheet [css]) classes are available that allow the formatting of specific sections within the SPL instance. For example, css allows a paragraph to appear as a 'Black box' when displayed even though a specific 'black box' element or attribute value is not defined in the SPL schema. Formatting codes are included in SPL as the styleCode attribute in most narrative block elements (see Sec. 4.1.4, Formatting SPL).

An SPL document is used in this guide as a general term to refer to any SPL document; SPL instance is used to refer a specific SPL document, e.g., the SPL Release 1 singulair example available at http://www.fda.gov/oc/datacouncil/spl.html. For the purposes of the implementation guide, these terms are used synonymously.

Note: the previously released stylesheet for SPL Release 1 will not work for FDA SPL '2a' documents (i.e., instances). Please check http://www.fda.gov/oc/datacouncil/spl.html. for information regarding the availability of the 'FDA SPL 2a' stylesheet.

[5] Rendition, display, and presentation are all used synonymously in this section to refer to the display of SPL on an output device, e.g., a web browser.

6 Technical details regarding the HL7 standard stylesheet are provided in Section 7.2 SPL Standard Stylesheet and FDA Implementation of Stylesheets, although how the stylesheet is used to display SPL is discussed in Section 4.1.4: Formatting SPL.

2.2 The SPL Document

An SPL document consists conceptually of two sections:

- Header
- Body

The body comprises the content of labeling, with two representations of the content:

- Labeling content (human readable text)[7]
- Data elements (machine processable content)[8]

This is illustrated below:

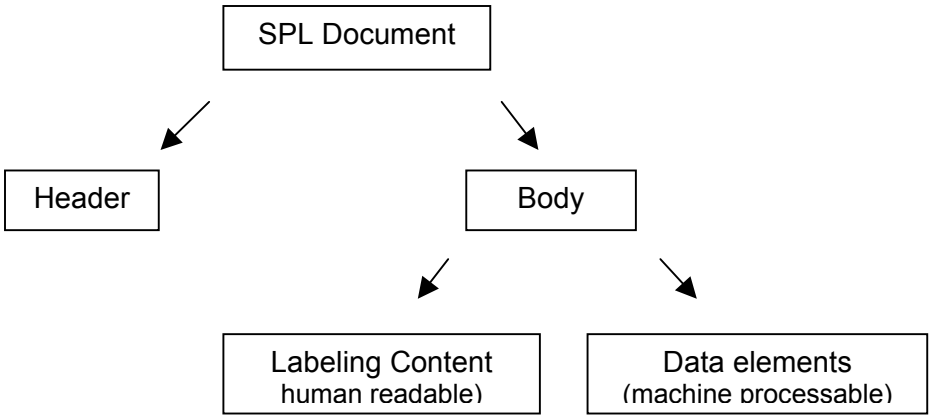


Figure 1. Conceptual SPL Structure

The following sections address construction of the header and body. The two parts of the body, the labeling content and the data elements, are considered separately in Sections 4.1 and 5, respectively.

An SPL document (as does all XML documents) must include a special section at the start of the document with processing instructions and the root element. Although these are XML structures and are not unique to the SPL header, for convenience they are discussed in the Creating the SPL Header section below since they will always be the initial part of an SPL document.

3. Creating the SPL Header

3.1 Processing Instructions and the Root Element

All XML documents, including SPL (which is a specific type of XML document), must include processing instructions and the root element[9]. The processing instructions at the start of SPL and the root element must be identical for every document submitted to FDA and have the following form:

[7] The label content in SPL is also referred to as the narrative, the narrative block, or narrative text, reflecting its origin from the narrative text of the printed package insert.
[8] See Sec. 5, Creating the Drug Listing Data Elements Section.
[9] Processing instructions and the root element are discussed in Section 7.5. These are mandatory parts of an XML document; as such, they are not directly relevant to the discussion of the conceptual outline of an SPL document that follows. It should simply be noted that these lines must appear at the start of any SPL document submitted to FDA to be a

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="spl-1.0.xsl "?>
<document xmlns="urn:hl7-org:v3" xmlns:voc="urn:hl7-org:v3/voc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">[10]
```

Although this information appears at the start of each SPL document, it is conceptually separate from the SPL header (as discussed previously and explained further below). It may be best considered as a mandatory part of an XML document and as such is not included in descriptions of the SPL header.[11]

3.2 SPL Header Elements

The header contains information about the document (SPL metadata). It is similar to the type of information that would be contained in the 'properties' box of a word processing document or in the information that a document management system would use for identifying a document.

The header section contains the following elements following the <document> element.[12,13] With the exception of the <title> element in the header, none of the elements in the header that are optional in the SPL schema are necessary in SPL documents submitted to FDA. These fields may be used by the author but the information will not be processed by FDA.

valid SPL document (see Section 7.5, XML Primer for a discussion in of XML/SPL validity). Note that the start of an SPL document may not begin exactly as in the text since comments may be interspersed.

[10] The following general conventions are followed in this document (with occasional exceptions). Element names in examples are in brown and usually in italics in the text, e.g., <car> in an example and *car* when discussed in the text. The left and right angle brackets for an element are in blue as in <car>. Attribute names are in red, e.g., <person age="20"> and attribute values in black. This general convention may be expanded on or altered in certain complex examples. For more information regarding elements and attributes, see Appendix 7.5 - XML Primer. All elements in SPL begin in lower case and follow a camelCase convention (e.g., <activeIngredient>).

[11] This specific code may change, particularly as the xsl file is updated to later versions. The most current recommendations for the processing instruction/root element syntax is available at www.fda.gov/oc/datacouncil/spl/

[12] Please refer to the SPL normative standard at http://www.hl7.org/Special/committees/rcrim/docs.cfm for a complete description of all SPL header elements and attributes.

[13] The <document> element is the root element of an SPL document.

Table 1. SPL Header Elements[a]

Element	SPL Schema Req.	FDA Req.	Comment	Examples
id	Yes	Yes	<id> is a globally unique identifier for the specific document instance and will differ for every regulatory submission. Information regarding the creation of an id for a specific SPL instance (and for other parts of SPL) is discussed in Sections 4.1.3.1 and 7.8	<id root="35F683E0-94C2-47FE-8925-43D170787B5D"/>
code	Yes	Yes	<code> represents the LOINC code for human prescription drug labeling[14]. This line should be identical in all SPL submitted to FDA at this time. As implementation of SPL is expanded to other type of product labeling (e.g., OTC labeling), other codes will be available. It is important to note that for the <code> element and in all subsequent elements where codeSystemName is used, this attribute is optional since the codeSystem is determined by the value for the codeSystem. Similarly, the displayName is unnecessary since the value for the displayName is contained in the code value. These attributes are only for human readability and are otherwise unnecessary.	<code code="34391-3" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" displayName="Human prescription drug label"/>
title	No	Yes	<title> should correspond exactly to the title string on the package insert with the exception that trademark and registered trademarks should not be included in the character string. The <title> element is the only header element rendered by the standard stylesheet. The SPL release 2 schema permits tags within the <title> element for multiline titles. <content …>, <sup>, and <sub> tags are also suppotted within <ttile> for formatting the title.	Example 1: <title>GEMZAR® (GEMCITABINE HCL) FOR INJECTION</title>
effectiveTime	Yes	Yes	This element is required by the SPL schema (and therefore must be present) but this value is not used from the actual submission. This value must use the HL7 TS data type; although different formats can be used, yyyymmdd is recommended. It is expected that FDA will populate this field with the approval date/time after a labeling change is approved or accepted (e.g., CBE) by FDA and will be present when SPL documents are exported from FDA to the National library of Medicine	<effectiveTime value="20050106"/>
availabilityTime	No	No	Not used at present. If included, it will be ignored by the FDA receiving system. If this element is used it should have an HL7 time stamp format.	

[14] LOINC is one of the coding systems that have been adopted by HL7 for use in SPL and other HL7 standards. LOINC codes to be used with SPL are listed in section 7.6, 'LOINC codes for SPL'. For a complete description of the LOINC database, see http://www.LOINC.org.

confidentialityCode	No	No	All FDA submissions are considered confidential by definition. If used, this code must be taken from the HL7 value set but is not used internally by FDA.	
languageCode	No	No	All submissions to FDA are required to be in English rendering this field unnecessary. If used for internal systems this code is taken from the HL7 value set for this element.	
setID	No	Yes	<set ID> will be a unique identifier for the document that will remain constant through all versions/revisions of the document. The value for the <setID> root attribute must be a GUID (described below) FDA will use this information to identify and process changes to a particular product label.	<setID root=" 372CB899-A37A-4898-8E91-B96B849C3673"/>
versionNumber	No	No	<versionNumber> will identify a version of the document; the combination of <setID> and <versionID> will be unique for each approved version of a product label. This field will be added by FDA when SPL is exported when FDA to NLM, and will be incremented for each new approved version of labeling. This field should not be included (or should be blank) when SPL is submitted to FDA.	
author	No	No	<author> is a complex element that is not yet required by FDA. If this element is included in an SPL, the <author><assignedEntity><representedOrganization> construct should be used to identify the sponsor of the labeling. In most cases this should correspond to the sponsor's identification of the FDA 356h form.The complete description of the complexType author element is available in the SPL normative standard. If submitted with SPL, the format should be use of the <addr> child elements <streetAddressLine>, <city>, <state>, <postalCode>, and <country>. . If the element is used, a phone number and fax number should be included using the <telecom> element.. If a sponsor wishes to include an optional timestamp for the date a SPL document is submitted to FDA, it should be included with this element as the <time value=""/> child of author, e.g., <author><time value="20050325"/><assignedEntity>…. The contents of this field will not be exported with SPL when FDA releases an approved SPL at this time. The use of this element may be redefined in the future although the structure will be unchanged.	<author> <assignedEntity> <representedOrganization> <name>My Drug Company</name> <addr> <streetAddressLine>1234 MyOffice Place</streetAddressLine> <city>Rockville</city> <state>MD</state> <postalCode>20853 </postalCode> <country>US</country> </addr> <telecom value="tel:(123)456-1212"/> <telecom value="fax:(123)789-1213"/> </representedOrganization> </assignedEntity> </author>

legalAuthenticator	No	No[b]	<legalAuthenticator> is a complex element not used by FDA at present; contents of this element will be ignored by FDA. The complete description of the complexType <legalAuthenticator> element is available in the SPL ballot.	
verifier	No	No[b]	<verifier> is a complex element not used by FDA at present; contents of this element will be ignored by FDA. The complete description of the complexType <verifier> element is available in the SPL ballot.	
relatedDocument	No	No[b]	<relatedDocument> is a complex element permitting reference to other SPL documents through the setID and versionNumber child elements. relatedDocument is not used by FDA at present; contents of this element will be ignored by FDA. The complete description of the complexType <relatedDocument> element is available in the SPL ballot.	

[a] If additional requirements are identified in future, this document will be updated to include them

[b] These elements are not required by FDA at this time, and if included the information will not be processed. The specifications for these fields, if these fields are implemented, will be published when the full Electronic Labeling Information Processing System (ELIPS) is implemented at FDA.

The following is an example of the SPL header (with the root element) as it would appear in a FDA submission. (Please note that the <author> element is not required at this time.)

```
<Document xmlns="urn:hl7-org:v3" xmlns:voc="urn:hl7-org:v3/voc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id root="81E32825-5BC8-46EB-8043-AE607B3819FA"/>
    <code code="34391-3" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
    displayName="Human prescription drug label"/>
  <title>GEMZAR<br/>(GEMCITABINE HCL) FOR INJECTION</title>
  <effectiveTime value="20050106"/>
  <setID root="372CB899-A37A-4898-8E91-B96B849C3673"/>
  <author>
      <assignedEntity>
          <representedOrganization>
              <name>My Drug Company</name>
              <addr>
                  <streetAddressLine>1234 MyOffice Place</streetAddressLine>
                  <city>Rockville</city>
                  <state>MD</state>
                  <postalCode>20853</postalCode>
                  <country>US</country>
              </addr>
              <telecom value="tel:(123)456-1212"/>
              <telecom value="fax:(123)789-1213"/>
          </representedOrganization>
      </assignedEntity>
  </author>
```

Note: For the <code> element and in all subsequent elements where codeSystemName is used, the codeSystemName attribute is optional since the codeSystem is determined by the value for the codeSystem attribute, not by the value of codeSystemName. Similarly, the displayName is unnecessary since the value for the displayName is contained in the actual code value. These attributes are only for human readability and are otherwise unnecessary. However, at this time it is recommended these attributes be included for confirming that the appropriate code has been entered.

4. Creating the SPL Body

In addition to SPL header information, the <document> element contains a required <component> which contains the <structuredBody> element. The <component><structuredBody> tags enclose the body of the SPL document; the body consists of the human readable content of labeling (i.e., the narrative text) plus structured data elements intended for machine processing (currently limited to specific drug listing information regarding the drug product, e.g., the active ingredients).[15]

The primary "building blocks" for the body of the document are sections. 'Sections' of the label content (or 'sections' of the narrative) represent related information; for example, each major 'section' of the printed labeling (e.g., Description, Indications and usage, Warnings) should be marked as a section in SPL. A section may contain sections, i.e., there may be sub-sections. *In every case, a section contains paragraphs of information that are related and belong together.* For example, several paragraphs discussing a specific precaution would be a sub-section within the larger 'Precautions' section. This is discussed further below.

4.1 Sections

In the SPL schema, the <structuredBody> sequence contains multiple <component>s, and within <component>s, each contains a <section>. This is illustrated below. The example is not valid SPL code and is used only to illustrate the structure of SPL[16]:

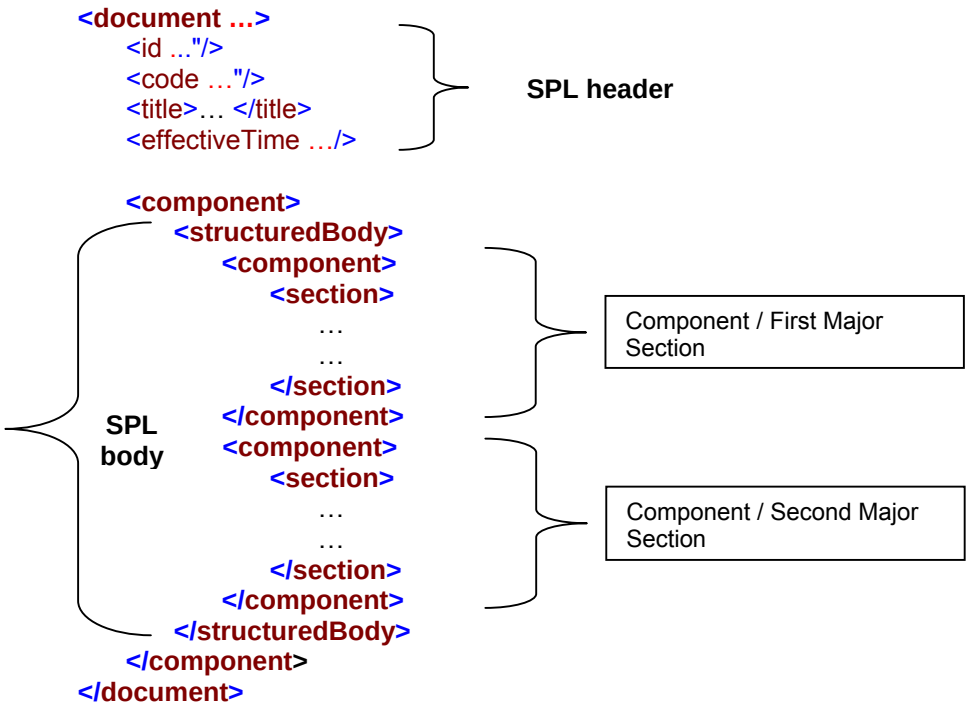


Figure 2. Example of SPL structure for structuredBody and sections within a structuredBody.

[15] Please refer to the SPL standard at http://www.hl7.org/Special/committees/rcrim/docs.cfm for a complete description of all SPL body elements and attributes.

[16] Although the option exists for a nonXMLBody in the SPL schema (i.e., <nonXMLBody> instead of <structuredBody>), all SPL submissions to FDA must use the <structuredBody> structure/element after the header elements.

Sections are used to aggregate paragraphs into logical groupings. For the FDA implementation of SPL, some of the <section>s are the major sections of labeling as defined by the labeling regulations in 21 CFR 201.56 and 57 (e.g., Indications and Usage) and are defined by LOINC codes; others are uncoded subsections that may or may not be identified with a title.

There should only be one section coded for each of the label section as defined in 201.57, i.e., there should only be one SPL section coded as Clinical Pharmacology, one section coded as Indications and Usage, etc., with the exception of coding for black box warnings.[17] Guidance regarding labels that cannot be coded appropriately in SPL using the one section restriction (e.g., labels that contain multiple, separated 'Indications and Usage' sections) will be available in an SPL FAQ currently under development..

An example of a section identified by a LOINC code with a sub-section not identified by a LOINC code:

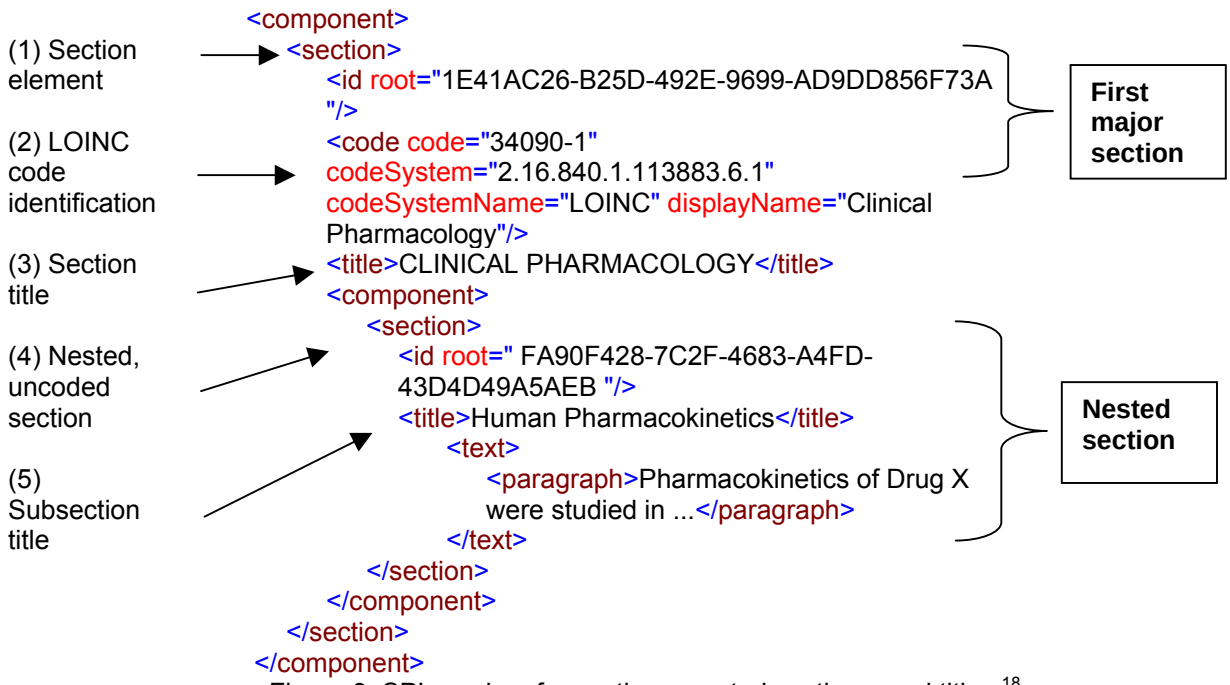


Figure 3. SPL markup for sections, nested sections, and titles.[18]

The SPL standard does not dictate the order of the sections; it merely provides a mechanism for identifying them. Therefore, it is important to note that the order in which sections are added to an SPL document is the order the sections will appear when displayed (rendered) using the standard stylesheet. Standard rendering of the content of SPL (see Formatting and Stylesheets below) results in display of sections in the order in which they appear in the source XML document. The required section order and section nomenclature are specified in FDA regulations.

A <section> may also contain sub-elements or metadata that uniquely identify and classify the section, similar to what is used to identify the document in the SPL header. As shown in Figure 3, each section has

[17] Black box warnings ("boxed warnings") should always be identified by the appropriate LOINC code (34066-1) and enclosed in a separate <component><section>. Whether the section stands alone or is nested within another section will depend on the context in which the boxed warning is included in the labeling.

[18] This structure may be counterintuitive, i.e., why a <component><section> tag is always needed for non-nested sections rather than <section> alone. The model-based derivation of the SPL schema from the overall HL7 Reference Information Model mandates this element. The author only need note that for non-nested sections, <component><section> should be used. (Similarly, <component> preceding <structuredBody> is mandated at the start of the body section although the <component> tag may appear unnecessary.)

a unique identifier (<id>), may be identified semantically by a LOINC code (i.e., the <code> element), and may contain a <title>. These are also described further below.[19]

The human readable content of labeling is contained within the <text> element in <section>s.[20] It should be noted that in all cases the structured narrative contained in SPL must match the narrative text (i.e., the content of labeling) as it exists in the printed final product labeling.

4.1.1 Nesting of Sections and Subsections

<section>s can nest to form sub<section>s. The schema for subsections in SPL requires that the nested <section> tag first be nested inside a <component> tag, as illustrated in the Figure 4 above.

The <component> element is used for nesting a section within another section. The following illustrates the method for creating nested sections (using non-valid code for illustrative purposes):[21]

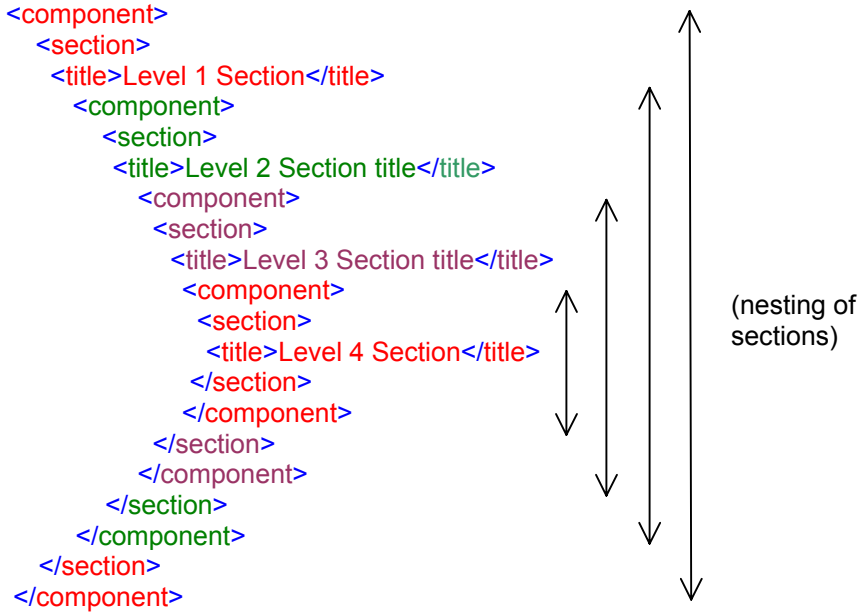


Figure 4. Use of <component> and <section> markup to nest sections in SPL.

4.1.2 Best Practices for Creating Sections

Best practice for markup of the label content (i.e., narrative) involves tagging for content rather than appearance.

- Multiple paragraphs are related by the use of nested sections: If information that belongs together is captured within a section, applications can identify the information as related and present this to

[19] A section may also have an id attribute, e.g., <section ID="1E41AC26-B25D-492E-9699-AD9DD856F73A">. This is discussed further below.
[20] A separate NarrativeBlock schema referenced by the main SPL schema describes the content model for <text>. This is described further below.
[21] Similar to first-level sections, the nesting of sections by <component><section> tagging may seem unnecessary, but is mandated by HL7 methodology.

users in customized applications that use SPL. Future use of SPL will make extensive use of sections as identifying related information. [22]

- Use of the <title> element is the preferred method for capturing string text that appears as a heading in current labeling documents.
- Use of either a caption on a paragraph or special formatting on a string of text (e.g., italics) to obtain the appearance of a heading is not acceptable.
- An ID can be included as an attribute to the <section> element, e.g., <section ID="4E41AC26-B25D-492E-9699-AD9DD856F73A"> if the section is to serve as the target of a <linkHtml> element. Linking to the ID attribute of a section allows the link to 'reference' the section entirely, e.g., for retrieval of a whole section in a non-browser interface.

It is possible to represent the following example using acceptable markup or unacceptable markup:

CLINICAL PHARMACOLOGY
Human Pharmacokinetics—Pharmacokinetics of Drug X were studied in .....

In patients with renal function impairment .....

Animal studies—In monkey studies over a two-year period ……..

The following example is *acceptable* markup format when there are subsections (i.e., explicit or implicit sub-headings) within a major section:

```
<component>
   <section>
      <id root="2EFE2C58-A9EB-4689-BD05-443C534D7D93"/>
      <code code="34090-1" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
      displayName="Clinical pharmacology"/>
      <title>CLINICAL PHARMACOLOGY</title>
      <component>
         <section>
            <id root="5708BCA0-8189-4331-83BA-08C7BDBD5C03"/>
            <title>Human Pharmacokinetics</title>
               <text>
                  <paragraph>Pharmacokinetics of Drug X were studied in ...</paragraph>
               </text>
         </section>
      </component>
      <component>
         <section>
            <id root=" CAC6A763-D3FB-42B2-9187-570BFA1CBE39"/>
            <title>Animal studies</title>
               <text>
                  <paragraph> In monkeys studied over a two-year period...</paragraph>
               </text>
         </section>
      </component>
```

[22] For example, in an adverse events section without a subheading, 3 paragraphs may describe the renal toxicity of the drug. These paragraphs should be captured in a single, separate section reflecting that this information forms a separate 'semantic' unit. This would be even more apparent if a subheading were present in the original text identifying the information as a separate and related.

```
    </section>
  </component>
```

The following is the same content as in the above example, but the markup is *not* acceptable. The markup format does not clearly delineate the relationships between the section, subsections, and paragraphs of the labeling content. The major sections have been lost. This markup format is not an acceptable SPL FDA submission.

```
<component>
    <section>
        <id root="51A61815-06CE-47CA-A2D2-ECB2F24EFA44"/>
        <code code="34090-1" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
       displayName="Clinical pharmacology"/>
        <title>CLINICAL PHARMACOLOGY</title>
        <text>
            <paragraph><content emphasis=”italics”>Human Pharmacokinetics</content> of Drug X
              were studied in ...</paragraph>
            <paragraph><content emphasis=”italics”>Animal studies</content> In monkeys studied
              over a two-year period ...</paragraph>
        </text>
    </section>
</component>
```

Both examples are valid against the SPL schema; however, the latter would not be acceptable for SPL FDA submissions. In the former example, if there were sections that identified additional granular units (e.g., <paragraph>In patients with renal function impairment ...</paragraph>, then these paragraphs should also be separated into separate (sub)sections.

4.1.3 <section> Elements

The <section> element can contain the child elements described in Table 2 below. Each element is optional under the schema except for the <id> element. All fields may be used by the author but only the <id>, <code>, <title>, and <text> elements will be processed by FDA at this time. Values for elements in each <section> that are similarly named to elements in the SPL header (other than the <ID>, <code>, and <title> elements) inherit the values of the header if they are not specified in the section but were included in the header. For example, if an optional element were included in the header (e.g., <author>), by default the value for <author> in each section would be the same as the header value if a value for author is not explicitly included in a specific section.

4.1.3.1 <id> elements

The <id> element is present in each section and in the header of the SPL document. The <id> tag takes the form <id root="…."> where the value for root must be a Globally Unique Identifier (or GUID), also known as a Universally Unique Identifier (or UUID)[23]. Each <id> root value must have a unique GUID different from every other GUID that exists *anywhere*. This mandates that GUIDs cannot be generated manually, since this could not insure that a specific GUID would be different from all other GUIDs that exist.

Multiple shareware/freeware computer programs exist that generate GUIDs automatically[24]. GUIDs are 128 bit integer values, or in hexadecimal, 32 hexadecimal digits. Examples are: 1C35F85F-9DE8-41CB-92EA-AC343157A935 and E470F428-7C2F-4683-A4FD-43D4D49A5AEB.

[23] UUID and GUID are used synonymously in this document. See Section 7.9.6 for additional information regarding GUIDs.
[24] Numerous freeware computer programs exist that generate GUID values, e.g., GuidGen at http://www.csc.calpoly.edu/~bfriesen/software/console.shtml or GUIDgen at http://www.microsoft.com.

Please also note that the <id> element is separate from the ID attribute that may exist on a <section> element, e.g., <section ID="E470F428-7C2F-4683-A4FD-43D4D49A5AEB">

4.1.3.2 <code> elements

Sections that represent regulatorily mandated labeling sections (e.g, INDICATIONS AND USAGE, WARNINGS, etc.) must have the appropriate <code> element following the <section> element. <code> contains the attribute values for the LOINC code that matches the specified section. <code> tags take the form <code code="…." codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" displayName="……."/>, e.g., <code code="34067-9" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" displayName="Indications and Usage"/>.. The codeSystem and codeSystemName attributes are always codeSystem="2.16.840.1.113883.6.1" and codeSystemName="LOINC". (The LOINC code system is the only FDA-acceptable code system for this attribute.) The code will be the LOINC code for the specific section as in this example. The complete list of LOINC codes for use with SPL is listed in section 7.6, LOINC codes for SPL.

If a LOINC code is not available (i.e., for subsections not mandated by FDA regulations), the <code> element should not be included in the section, i.e., <section><id><text>….. should be used.

The available LOINC code set for Human Prescription Drug Product sections has been determined by the label (package insert) sections defined in the FDA regulations. The LOINC code for Human Prescription Drug label is 34391-3. Other LOINC codes for <section><code>s available as of October 2004 are in Table 17: LOINC Codes in SPL. When submitting SPL documents to FDA, use of these codes is required for sections mandated by regulation.

4.1.3.3 <text> elements

The <text> element in a <section> can contain actual text (also known in XML as PCDATA[25]) and the following elements:

- paragraphs (<paragraph>)
- lists (<list>)
- tables (<tables>)
- images <renderMultimedia>)

The following elements are also permitted as children of the <text> element, but it is recommended they only be used as children of the <paragraph> element or within <table>s and <list>s.[26]

- superscripts (<sup>)
- subscripts (<sub>)
- links (<linkHtml>)
- revision of content (<content revised="…">)
- line breaks (<br>)
- footnotes (<footnote>)
- footnote references (<footnoteref>)

[25] PCDATA is normal character data, i.e., the normal content of labeling. Certain characters cannot be used (e.g., <), but essentially this will be information as you might normally enter in a word processor. 'Special' characters that cannot be used directly are entered as entity references or Unicode; this is discussed in section 4.1.4.3

[26] For example, although '<text><linkHtml href="http://www.fda.gov">this is permitted³</text>', is permitted, it is recommended that actual content be contained within one of the former elements (i.e., paragraph, table, etc.), e.g., '<text><paragraph><linkHtml href="www.fda.gov">this is enclosed in a paragraph tag³ </paragraph></text>'

The <text> element contains labeling content, i.e., the human readable text content of SPL that is displayed (rendered). It is recommended, however, that actual content, be contained within a <paragraph>, <table>, or <list>.[26]

Within paragraphs, text may be enclosed by […] (superscripts), _… (subscripts) for formatting. Footnotes, footnote references, links, and line breaks can also be identified by the appropriate tags. The content element can be used to indicate document revisions by the asscoiated "revised" attribute, which has possible values 'insert' and 'delete'. For example,

```
<paragraph>The <content revised="delete"> quick brown </content> fox
<content revised="delete">jumped</content><content revised="insert">leapt</content> over the
lazy <content revised="insert">jet</content> black frog.
</paragraph>
```

could be rendered as:

The ~~quick brown~~ fox ~~jumped~~ leapt over the lazy jet black frog.

Inline images may be included in the content of labeling via the <renderMultiMedia> tag. This tag may be used as a direct child of <text> for 'separate' images or as a child of <paragraph> for inline images. The <renderMultimedia> element is described under Images below (see Sec. 4.1.6).

Although under the SPL schema the <content> tag potentially could be used for multiple purposes, it should only be used to mark revisions to text (see Sec. 4.1.4.2) or as a potential anchor for links (i.e., the ID attribute of a <content> tag could be the link target.

A representation of the <text> element is reproduced below, followed by a description of all child elements of the <section> element.

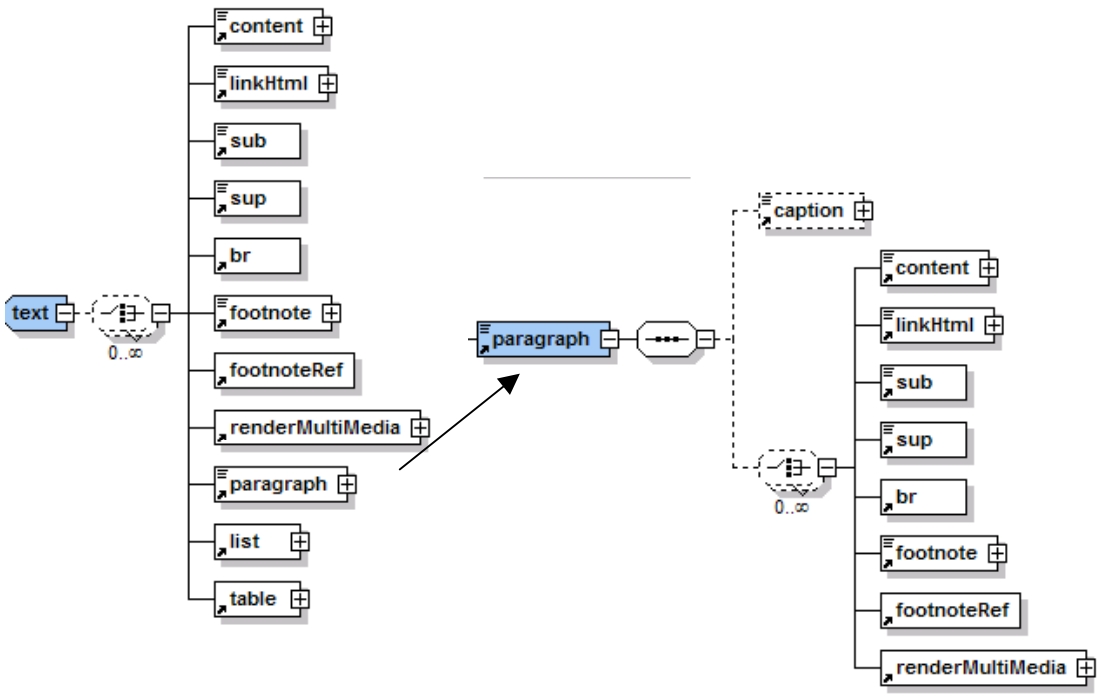


Figure 5: Schema for <text> and <paragraph> elements.

4.1.3.4 <subjectOf><comment> elements

The <subjectOf> element is used to insert comments/annotations into an SPL document. (The standard XML comment construct, i.e., <!-- …> should NOT be used in SPL.) The <comment> is separate from the actual narrative block content and refers to the content via a reference. The structure of a comment is as follows:

```
<subjectOf>
    <comment>
    <text>
    <statusCode>
    <effectiveTime>
    <author>
    <subject>
      <textFragmentReference>
        <text>
          <reference url="">
    <sequel>
      <comment>
</subjectOf>
```

As an example:

```
<section>
  …….
  <text>
```

```
    <paragraph> this is narrative text describing something <content ID=”cmt01”>that
    may be important to comment on</content> continuing paragraph</paragraph>
  </text>
  ……..
  <subjectOf>
    <comment>
      <text>This is not an important point and should be deleted</text>
      <statusCode code="active"/>
      <effectiveTime value="20050106"/>
      <author>
        <assignedEntity>
          <assignedPerson>
            <name>Joe Doe</name>
          </assignedPerson>
        </assignedEntity>
      </author>
      <subject>
        <textFragmentReference>
          <text>
            <reference url="#cmt01"/>
          </text>
        </textFragmentReference>
      </subject>
    </comment>
  </subjectOf>
  …….
</section>
```

Acceptable values for statusCode code attribute are code=“active” or code=“complete”. Nesting of comments is permitted via the <sequel> tag where additional <comment>s can be nested below the original comment.

At the present time the <subjectOf> construct is ignored when SPL is received by FDA; however, it is expected that in the future ELIPS will use this construct to support label annotations by sponsors and the communication of labeling comments between sponsors and FDA.

4.1.3.5 Listing of all <section> elements

Table 2. SPL Elements within the <section> Element[a]

Element	Sch./FDA Req.[a]	Comment	Examples
id	Yes/Yes	<id> is a globally unique identifier for the specific section instance. See Section 4.1.3.1 for more information regarding creation of an id for an SPL section. This is separate from an ID attribute for the section itself.	<id root="2 F33776B3-2DC8-435B-856B-444DD69F6CD7"/>
code	No/Yes[b]	<code> represents the LOINC code for the section (e.g., Indications and Usage). The list of currently available US-labeling document section codes that have been assigned by LOINC is included in the searchable LOINC database (http://www.loinc.org) or see Table 17 below. When no code for a section is available, a local code may be used but will be ignored by FDA. The displayName in the code is for information purposes only – it is not used to generate a title for a section in the rendered document. For additional material that is part of the content of labeling after the How Supplied Section, e.g., if a Patient Package Insert or MediGuide is included as the final part of a package insert, the LOINC code for Supplemental Patient Information (38056-8) should be used. Within a section identified by this LOINC codes, subsections that further identify the information as either a Patient Package Insert (PPI) or MedGuide can be used. Codes for these sections are listed in see Table 17; however, codes for the MedGuide or PPI must always be used as subsections of the Supplemental Patient Information section.	<code code="34067-9" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" displayName="Indications and Usage"/>
title	No/Yes[c]	The major sections in the labeling document must have titles (the appropriate titles are defined by FDA regulations). The title of a section is rendered from the content of the <title> tag by the standard stylesheet. If the <title> tag is not populated, then no title will be displayed. The title is NOT rendered from the value of displayName attribute if a <code> element is present. Not every section will have a title; however even in the absence of a title, paragraphs should be grouped into separate sections based on relationships between the content (see 4.1.2). Titles should be included whenever they are present in a printed document from which label content has been converted to SPL. Sections and their titles may be nested, resulting in an implied hierarchy that is rendered appropriately in the standard stylesheet. For more information, see the “component” entry, below.	<title>INDICATIONS AND USAGE</title>

Element	Sch./FDA Req.[a]	Comment	Examples
text	No/ Yes[d]	The human readable content of labeling (the narrative) is contained within the <text> element. See below for additional discussion of the <text> content model.	<text><paragraph> This drug exhibits analgesic and ….. </paragraph></text>
effectiveTime	No/No	<effectiveTime> contains a timestamp for when the section was written. Format is timestamp format, yyyymmdd (4 digit year/2 digit month/2digit day). This field will be populated by FDA when an approved SPL is released by FDA to reflect the specific time a section was accepted. At the time of first approval this will be equal for all sections, but will then reflect updated times for specific sections if labels are revised.	
confidentialityCode	No/No	This code is used to override the document confidentiality code in the SPL header if the confidentiality level for the section is different. If used, this code is taken from the HL7 value set. There are no plans to use this element at this time.	
languageCode	No/No	This code is used to override the document language code in the SPL header if the language for the section is different. If used, this code is taken from the HL7 value set. There are no plans to use this element at this time.	
author	No/No	This element is used to override the author of the document (identified in the SPL header) if the author for the section is different. <author> is a complex element not used by FDA at present. The complete description of the complexType author element is available in the SPL specification.	
component	No/No	<component> is used to link sections to sections nested within them (see Figure 3 for levels of nesting available in SPL). Rendering of the titles for nested sections is set in the stylesheet and depends on the level of nesting. See Section 4.1.3.6.2 for more information.	<section> <id root="192D3B70-0457-4F92-AA4A-8B30928A32A9"/> <code code="34090-1" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" displayName="Clinical Pharmacology"/> <title>CLINICAL PHARMACOLOGY</title> <component> <section> <id root="192D3B70-0457-4F92-AA4A-8B30928A32A9"/> <title>Human Pharmacokinetics</title> <text> <paragraph>Pharmacokinetics of Drug X were…</paragraph> </text> </section> </component> </section>

Element	Sch./FDA Req.[a]	Comment	Examples
replacementOf	No/No	<replacementOf> is a complex element permitting reference to previous versions of a section through the <id> child of <sectionReplaced>. <replacementOf> is not used by FDA at present and contents of this element will be ignored by FDA. When the full ELIPS system is completed additional specifications for the use of this selement may be published. The complete description of the complexType <replacementOf> element is available in the SPL specification.	
subjectOf	No/No	<subjectOf><comment> is used to annotate narrative text. It is not used at present but is expected to support annotation of labeling and communication between sponsors and FDA when implemented by ELIPS. See Section 4.1.3.4	
subject	No/Yes	<subject> is used to contain the data elements for describing the drug listing data elemetnsf in SPL. See Section 5 for a full description.	
excerpt	No/No	<excerpt> is included in the SPL schema in anticipation of the physicians labeling rule. It contains the children <highlight> and <text> which are used to contain a text synopsis of a narrative section. This element should not be used at present.	

[a] If additional requirements are identified in future, this document will be updated to include them. 'Sch./FDA' required refers to whether the element is required by the schema and/or FDA. Elements that are required by the schema must be present although the information may not be used by FDA.
[b] <code> is required for sections that are mandated by regulation. <code> is not required for other sections
[c] Titles are required only for sections mandated in FDA regulations, but may be used for other sections as appropriate. Titles for subsections are rendered appropriate to their level of nesting.
[d] If a section consists only of nested sections, this tag is not required; however, it is required if any text in that section is to be rendered.

4.1.3.6 Sample Section Markup

4.1.3.6.1 Isolated Major Section

```
<!-- - - - - - - - - - -  Sample Contraindications section  - - - - - - - - - - - - - - - - - - - - - - -->
  <component>
    <section>
      <id root="02128CFE-34DB-428A-B92F-A155103C354D"/>
      <code code="34070-3" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
     displayName="Contraindications"/>
      <title>CONTRAINDICATIONS</title>
      <text>
        <paragraph>DrugX is contraindicated in those patients with a known hypersensitivity to
        the drug.
        </paragraph>
      </text>
    </section>
  </component>
```

4.1.3.6.2 Major Section (with LOINC code) and subsection (without LOINC code)

```
<!-- - - - - - - - - - -  Sample Clinical Pharmacology section- - - - - - - - - - - - - - - - - - - - -->
  <component>
```

```
<section>
    <id root="5031A439-E76A-4FEB-827D-9AC5A758076A "/>
    <code code="34090-1" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
   displayName="Clinical pharmacology"/>
    <title>CLINICAL PHARMACOLOGY</title>
    <component>
        <section>
            <id root="51A61815-06CE-47CA-A2D2-EFB2F24EFA44"/>
            <title>Human Pharmacokinetics</title>
                <text>
                    <paragraph>Pharmacokinetics of Drug X were studied in . .</paragraph>
                </text>
        </section>
    </component>
</section>
</component>
```

4.1.4 Formatting SPL

This section discusses several aspects broadly defined as 'Formatting SPL', including (a) use of the <styleCode> element for certain formatting options, (b) font effects (bold, underline, italics), (c) symbols and special characters (Unicode), (d) footnotes, and (e) default and specialized lists. Tables are discussed separately in Section 4.1.5.below.

4.1.4.1 <styleCode> attribute

A major design goal of XML (and SPL) documents is to separate formatting from content; accordingly, the SPL schema contains minimal formatting features. However, the SPL standard also specifies that an SPL document should be human readable, and further specifies that a standard stylesheet be available for rendering SPL labeling in modern Web browsers.

Despite use of a stylesheet with an SPL document, there are certain aspects of the rendering of SPL that must be specified in the SPL source to insure that the content of labeling is formatted correctly when rendered. Examples of this are rules separating rows in a table into a section or the printed box that defines a black box.

To accomplish this, SPL includes the *styleCode* attribute on many narrative text elements to add formatting information; this attribute is used to select CSS classes at the time the SPL document is rendered by the standard stylesheet.

For example:

> <paragraph>The next snippet <content styleCode="bold italic”> will appear as bold italic</content> in the rendering</paragraph>

will be rendered as :

> The next snippet ***will appear as bold italic*** in the rendering.

There are multiple examples of the use of the *styleCode* attribute in the singulair example at www.fda.gov/oc/datacouncil/spl.html, where the attribute is used to insure that tables have appropriate rules and that footnotes are formatted correctly. The use of the SPL stylesheet is discussed in Appendix 7.2

4.1.4.2 Font Effects

As discussed earlier, the <content styleCode="" > element is the primary mechanism for including font effects in text. Currently bold, italics, and underline are suppoted. These are shown in Table 3 below.

Table 3. Font Effects

Attribute value for 'styleCode'	Recognized by FDA	Example Rendering	Examples
bold	Yes	**contraindicated**	<content styleCode="bold">contraindicated</content>
italics	Yes	*in vitro*	<content styleCode="italics">in vitro</content>
underline	Yes	fever	<content styleCode="underline">fever</content>

Note that combined font effects for bold, italics, and underline are permitted in SPL as shown in the following table:

Table 4. Multiple Font Effects

Desired Font Effect	Rendering	Examples
bold-italics	***contraindicated***	<content styleCode="bold italic">contraindicated</content>
italics-underline	*in vitro*	<content styleCode="italics underline">in vitro</content>
bold-underline	**fever**	<content styleCode="bold underline">fever</content>
bold-italics-underline	***pharmacology***	<content styleCode="bold italics underline"> pharmacology</content>

4.1.4.3 Symbols and Special Characters

Special characters can be included in narrative (i.e., the text content) and may be created in different ways. Simple superscripts and subscripts are accomplished with tagging included in the SPL schema, i.e., <sup> and <sub> tags. Unicode[27] character references are used for special characters.[28]. The Unicode value for common symbols (e.g., &#8482 for ™) are in Table 5. Symbols and Special Characters. Complete lists are available in the references cited in footnote 27.

Table 5. Symbols and Special Characters

[27] Complete information regarding Unicode is available at http://www.unicode.org and http://www.alanwood.net/unicode/index.html#links. A well indexed table of Greek and other special characters can be found at http://www.alanwood.net/demos/symbol.html#s0080.

[28] Unicode character set UTF-8 is the default character set in W3C XML Schemas

Symbol or Character	Solution	Sample Markup
mg/m^2 (superscript)	Tagging	mg/m²
• (e.g., $C_9H_{11}F_2N_3O_4$•HCl)	Tagging + Unicode	C₉H₁₁F₂N₃</sub>O₄•HCl)
© (copyright)	Unicode	&#169;
º (degree)	Unicode	&#176;
<	Entity	&lt;
≤	Unicode	&#8804;
— (m-dash)	Unicode	&#8212;
β (e.g., β-isomer)	Unicode	β-isomer
® (e.g., Registered 1996)	Unicode	Registered® 1996.
™ (e.g., Trademark™ 1998)	Unicode	Trademark™ 1998
± (plus-minus sign)	Unicode	&#177;
& (ampersand)	Entity	&amp;
†	Unicode	&#8224;
‡	Unicode	&#8225;
§	Unicode	&#167;
' (apostrophe)	Entitiy	&apos;

Unicode characters in SPL (and XML) are inserted as either &#dddd; where dddd is the unicode value (for decimal values) or &#xdddd; when hexadecimal values are used.

Native XML authoring tools automatically insert the proper Unicode value for special characters or symbols during the authoring process. However, because of differences in coding systems and software, some symbols or special characters may not carry over properly when copied and pasted from a word processing document into an XML document. If text is copied or otherwise converted from a word processing document into SPL-compliant XML, verify that special symbols and characters are properly encoded in the XML document.

As noted in Appendix 7.2, the font used in the standard stylesheet is a Unicode font (Arial Unicode MS), assuring that Unicode values in SPL content will be rendered correctly if viewed by a browser supporting this font. Also note that because XML (SPL) tags begin with the less than < symbol, use of this symbol in text content must be replaced by the XML entity <. For example, <paragraph>The mean for group 1 was < the mean for group 2.</paragraph>

4.1.4.4 Footnotes

The SPL schema includes a specific footnote element <footnote>. Footnotes are rendered automatically by the standard SPL stylesheet. <footnoteRef> is used to refer to an earlier footnote, e.g., <footnote ID=”testNote”>this is the footnote content</footnote>now more content, still more content, etc., this will place a footnote mark at the new location <footnoteRef IDREF=”testNote”>.

Footnotes are rendered by the default stylesheet mode, at present Arabic numbers (i.e., 1,2 3). Alternatively, a specific footnote style can be rendered using a styleCode (<footnote styleCode=”LittleRoman”>. Styllecode options are the same as for lists in the following sections. A special styleCode value (“FootnoteMarks”) is used within tables for the series * † ‡ § ¶ # ♠ ♥ ♦ ♣ and can be specified separately.

Table 6. Footnotes[29]

SPL Footnote Markup	Sample Footnote

[29] Note this example is illustrative; the current stylesheet identifies footnotes by numbers rather than letters.

<paragraph><footnote>28-day schedule -- Gemzar plus cisplatin: Gemzar 1000 mg/m² on Days 1, 8, and 15 and cisplatin 100 mg/m² on Day 1 every 28 days; Single-agent cisplatin: cisplatin 100 mg/m² on Day 1 every 28 days.</footnote> </paragraph>	a 28-day schedule – Gemzar plus cisplatin: Gemzar 1000 mg/m^2 on Days 1, 8, and 15 and cisplatin 100 mg/m^2 on Day 1 every 28 days; Single-agent cisplatin: cisplatin 100 mg/m^2 on Day 1 every 28 days.

4.1.4.5 Lists (Default and Specialized)

All lists are marked up using the <list> tag, and each item in a list is marked with an <item> tag. The 'listType' attribute identifies the list as ordered (numbered) or unordered (bulleted). The default numbering style or bullet type is not specified in the SPL schema; default numbering and bulleting are controlled by the stylesheet.

Table 7. Default Lists

Markup	Presentation via Standard Stylesheet
<list listType="ordered"> <item> Recommendations for the safe handling of parenteral antineoplastic drugs. NIH publication No. 83-2621. US Government Printing Office, Washington, DC 20402. </item> <item> Council on Scientific Affairs: Guidelines for handling parenteral antineoplastics. JAMA 1985;253:1590. </item> </list>	1. Recommendations for the safe handling of parenteral antineoplastic drugs. NIH publication No. 83-2621. US Government Printing Office, Washington, DC 20402. 2. Council on Scientific Affairs: Guidelines for handling parenteral antineoplastics. JAMA 1985:253:1590.
<list listType="unordered"> <item> Recommendations for the safe handling of parenteral antineoplastic drugs. NIH publication No. 83-2621. US Government Printing Office, Washington, DC 20402. </item> <item> Council on Scientific Affairs: Guidelines for handling parenteral antineoplastics. JAMA 1985;253:1590. </item> </list>	• Recommendations for the safe handling of parenteral antineoplastic drugs. NIH publication No. 83-2621. US Government Printing Office, Washington, DC 20402. • Council on Scientific Affairs: Guidelines for handling parenteral antineoplastics. JAMA 1985:253:1590.

Lists featuring a standard set of specialized markers (standard specialized lists) can be created using the styleCode attribute with the <list> element. Options available for ordered lists are:

- Arabic (List is ordered using Arabic numerals: 1, 2, 3)
- LittleRoman (List is ordered using little Roman numerals: i, ii, iii)
- BigRoman (List is ordered using big Roman numerals: I, II, III)

- LittleAlpha (List is order using little alpha characters: a, b, c)
- BigAlpha (List is ordered using big alpha characters: A, B, C)

For example:

```
<list listType="ordered" styleCode="LittleRoman">
```

For unordered lists the following options exist:

- Disc (List bullets are simple solid discs: ●)
- Circle (List bullets are hollow discs: ○)
- Square (List bullets are solid squares: ■)

The following table shows examples of the styleCode values available for ordered and unordered lists.

Table 8. Specialized Lists

Ordered Lists:

Specialized Type	styleCode Value	Sample Rendering
Arabic (same as default ordered list)	Arabic	1. Recommenc publication N 2. Council on S JAMA 1985;:
Lowercase Roman Numeral	LittleRoman	i. Recommendatior publication No. 8: ii. Council on Scien JAMA 1985;253:
Uppercase Roman Numeral	BigRoman	I. Recommendatior publication No. 8: II. Council on Scien JAMA 1985;253:
Lowercase Western Alphabetical	LittleAlpha	a. Recommendatior publication No. 8: b. Council on Scien JAMA 1985;253:
Uppercase Western Alphabetical	BigAlpha	A. Recommendatior publication No. 8 B. Council on Scien JAMA 1985;253:

Unordered Lists:

Specialized Type	styleCode Value	Sample Rendering
Disc (same as default unordered list)	Disc	• Recommendatior publication No. 8 • Council on Scient JAMA 1985;253:
Circle, hollow disc	Circle	o Recommendatior publication No. 8: o Council on Scien JAMA 1985;253:
Filled square	Square	▪ Recommendatior publication No. 8: ▪ Council on Scien JAMA 1985;253:

In addition to the standard specialized lists, user-defined characters are also permitted as markers by nesting <caption> within the <item> tag, as shown in the following table. Note that any character, XML entity, or Unicode symbol may be used in the <caption>, and that the <caption>s for each <item> are not restricted to the same character.

Table 9. User-defined Characters

SPL Markup	Sample Rendering
<list> <item><caption>@</caption>Recommendations for the safe handling of parenteral antineoplastic drugs. NIH publication No. 83-2621.US Government Printing Office, Washington, DC 20402.</item> <item><caption>@</caption>Council on Scientific Affairs: Guidelines for handling parenteral antineoplastics. JAMA 1985;253:1590.</item> </list>	@ Recommendations for the s: publication No. 83-2621. US @ Council on Scientific Affairs: JAMA 1985;253:1590.

4.1.5 Tables

Table markup in SPL is similar to the XHTML table model for structural tagging (i.e., <tr>, <td>, etc.) but lacks much of the XHTML formatting and style markup.[30]

Tables should always be created with the full structure as shown below. The structure will display a standard typographical table, with rules between the caption (table title) and head, the head and body, and the body and footnote:[31]

Table 10. Sample Table

SPL Table Markup	Sample Table
<table> <caption>…</caption> <thead> <tr .../> </thead> <tfoot> <tr .../> </tfoot> <tbody> <tr .../> </tbody> </table>	**Table 1: Clearance and Half-Life for the "Typical" Patient**

Age	Clearance Men ($L/hr/m^2$)	Clearance Women ($L/hr/m^2$)	Half-Life[a] (min)	Half-Life[a] (min)
29	92.2	69.4	42	49
45	75.7	57	48	57
65	55.1	41.5	61	73
79	40.7	30.7	79	94

[a] Half-life for patients receiving a short infusion (<70 min).

Note that the SPL schema enforces the positioning of <tfoot> before <tbody>. <thead> and <tfoot> are optional elements in the SPL schema but <tbody> is required for an SPL table.

Always start with this standard table and test to see whether it is unambiguous and interpretable. Use the rule styleCodes specified below to modify the table only when absolutely necessary. SPL is used to communicate labeling *content* rather than the exact representation of drug information present in a typeset document, the table presentation in SPL is unlikely to exactly duplicate the presentation in word processed or typeset versions of the package insert.

[30] A complete description of the XHTML document model is available at http://www.w3.org/TR/xhtml-modularization/Overview.html. Specific defaults regarding the XHTML table model can be found at http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#s_tablemodule.

[31] Although this style is recommended, only the <tbody> element is required as a child of <table>.

4.1.5.1 Table Rules (Gridlines)

The SPL schema allows users to control rules for each cell by setting the styleCode attribute code on <td>, e.g., <td styleCode="Lrule">.

The rule codes are shown in the following table. Note that the control names are case sensitive.

Table 11. Optional Table Rules

Rule Placement	styleCode attribute	Appearance
Rule on left side of cell	Lrule	
Rule on right side of cell	Rrule	
Rule on top of cell	Toprule	
Rule on bottom of cell	Botrule	

No more than one rule control may be used in a cell, e.g., <td styleCode code="Botrule Lrule">Cell content </td>.

Rule control codes should be *used only when necessary for the interpretability of the table.* Use of these codes may result in overriding the default rules for tables.

Rather than setting the rule for each cell, table rules may also be controlled according to entire rows or columns by use of the styleCode attributes with <col>, <colgroup>, <thead>, <tfoot>, <tbody> and <tr> elements.

4.1.5.2 Horizontal rules

To make rowgroups appear with horizontal rules, use the styleCode attribute "Botrule" with the appropriate <tr> element. The *Botrule* value is rarely needed on the <td> element.

4.1.5.3 Vertical rules

The preferred method for using vertical rules is to define colgroups with *styleCode="Lrule"* or *"Rrule"* (or both). Only if this does not yield the desired vertical rule should you use the *Lrule* or *Rrule* code value with *styleCode* attributes on the <td> or <th> element itself.

Note: Vertical rules should be used *sparingly!* Good typography for tables means using few vertical rules.

4.1.5.4 Cell text alignment

Horizontal alignment: Similar to XHTML tables, the preferred method for aligning cell content within the margins is to use *<col align=".. "/>* in the <colgroup> element, though this can be used in the <colgroup> element as well. Valid values for *align* are “left”, “center”, “right”, “justify” (for full justification of contents within the cells), and “char” (for character alignment within the cells). Using the *<col align=".. "/>* markup ensures that the contents for all cells in the column share the same alignment.

Vertical alignment: Analogous

For cases in which the cell alignment must be different from other cells in the column, *align* is also available as an attribute on the other table elements, including <td>. Thus it is possible, but not recommended, to change the alignment for each cell. The following example, while not recommended, demonstrates the XML markup and the result of using the *align* attribute on <td>.

```
<tr>
        <td>Number of patients</td>
        <td align="right">260</td>
        <td align="left">262</td>
        <td>69</td>
        <td>66</td>
</tr>
 <tr>
        <td>&amp;#160;&amp;#160;Male</td>
        <td align="left">182</td>
        <td align="right">186</td>
        <td>64</td>
        <td>61</td>
</tr>

<tr>
        <td>&amp;#160;&amp;#160;Female</td>
        <td>78</td>
        <td>76</td>
        <td>5</td>
        <td>5</td>
</tr>
```

Number of patients	260	262	69	66
Male	182	186	64	61
Female	78	76	5	5

4.1.5.5 Footnotes

Markup for table footnote is the <tfoot> tag, as is common practice. Other than this, table footnotes are substantially the same as all footnotes in SPL. See the Footnotes section earlier in this document. A <footnote> within a table will be rendered at the bottom of the page and not with the table; <tfoot> should be used for table footnotes.

4.1.5.6 Table text spacing

In some instances, the use of a “tab” or text indentation is desirable in a given table cell, as in the “Trial” column of the following example:

Trial	28-day Schedule		21-day Schedule	
Treatment Arm	Drugapil/ Goodrug	Goodrug	Drugapil/ Goodrug	Goodrug/ Asistix
Number of patients	260	262	69	66
Male	182	186	64	61
Female	78	76	5	5
Median age, years	62	63	58	60
Range	36 to 88	35 to 79	33 to 76	35 to 75

Note that “Male,” “Female,” and “Range” appear offset from the cell margin. In an SPL document, this effect is achieved by using the nonbreaking space ( ) as if it were a “tab” space. As the following snippet of XML shows, two nonbreaking spaces were used to offset the word “Male” from the margin:

```
<td>&amp;#160;&amp;#160;Male</td>
```

The nonbreaking space can also be used to keep text in a table from breaking inappropriately due to browser resizing. To build upon the above example, if the user did not want “Number of patients” to wrap, he or she can use the nonbreaking space to ensure this. The following XML encoding

```
<td>Number&amp;#160;of&amp;#160;patients</td>
```

ensures that the text will not break when the browser window is resized.

Trial	28-day Schedule		21-day Schedule	
Treatment Arm	Drugapil/ Goodrug	Goodrug	Drugapil/ Goodrug	Goodrug/ Asistix
Number of patients	260	262	69	66
Male	182	186	64	61
Female	78	76	5	5

4.1.6 Images

The SPL schema uses <observationMedia> elements to identify graphic files to be rendered at the locations where they are referenced by <renderMultiMedia> elements in the <section>. In other words, an image in an SPL will be rendered wherever it is referenced by the renderMultimedia markup, no matter where the observationMedia markup appears. The referencedObject attribute of the renderMultiMedia element identifies the corresponding ObservationMedia instance by means of its ID identifier.

```
<renderMultiMedia referencedObject="MM1"/>
```

Per XML convention, the <observationMedia> element does not contain the graphic file, but instead points at the file. Additionally, ObservationMedia identifies the graphic media type (JPEG or GIF). Note also that observationMedia is always contained within a <component> element.

```
<component>
    <observationMedia ID="MM1">
```

```
            <value xsi:type="ED" mediaType="image/jpeg">
                <reference value="gemzar_structure.jpg"/>
            </value>
        </observationMedia>
    </component>
```

For example:

```
    …
<paragraph>
    <content styleCode="underline">Quality of Life (QOL)</content>:
    QOL was a secondary endpoint in both randomized studies. In the Gemzar plus
    cisplatin versus cisplatin study, QOL was measured using the FACT-L, which
    assessed physical, social, emotional and functional well-being, and lung cancer
    symptoms. In the study of Gemzar plus cisplatin versus etoposide plus cisplatin,
    QOL was measured using the EORTC QLQ-C30 and LC13, which assessed physical
    and psychological functioning and symptoms related to both lung cancer and
    its treatment. In both studies no significant differences were observed in
    QOL between the Gemzar plus cisplatin arm and the comparator arm.
</paragraph>
<renderMultiMedia referencedObject="MM1">
…
…
```

would display image MM1 defined by <observationMedia ID="MM1"> in the document as a block image.

The actual image file is enclosed by an <observationMedia> element, contained in any <component> in any section, e.g.,

```
<section>
……..
<text>…… </text>
<component>
    <observationMedia ID="MM2">
        <value xsi:type="ED" mediaType="image/jpeg">
            <reference value="gemzar_2.jpg"/>
        </value>
    </observationMedia>
</component>
</section>
```

4.1.6.1 Size and resolution

The SPL schema does not allow for resizing graphics or changing the resolution of graphics files. Thus, all images are rendered in the browser as-is, with all characteristics of the actual graphic file itself. To ensure that a graphic will appear as required, the graphic file should be edited to a dimension appropriate for its presentation within the browser. Also, graphics files should be saved at a resolution appropriate for screen viewing, not printing. One common appropriate resolution for screen graphics is 72 dpi.

4.1.6.2 File type

The file type for images should be appropriate for the intended use of the SPL document. For SPL graphic files to be viewed in a browser according to the standard stylesheet only JPEG and GIF files should be used; JPEG is the preferred format. Additional details regarding image formats and naming image files are included in Appendix 7.4.

4.1.6.3 Image placement

If an image is *inline* (i.e., should appear alongside text), insert the renderMultimedia tag in the text of a <paragraph> as appropriate. If an image is a *block image* (i.e., should appear in its own space), insert the renderMultimedia tag after the closing </paragraph> tag.

Note: Include any company or product logos as image files; do not attempt to recreate a logo with font effects in the XML document. The standard stylesheet controls the font family and effects for all text in an SPL-compliant package insert, so any font information in the XML will be lost.

4.1.7 Hypertext Links

SPL offers hypertext linking capabilities generally similar to those found in the HTML specification. Links are specified by the <linkHtml> construct, where the value for the href attribute of <linkHtml> (the target of the link) is the ID attribute value of a <section>, <paragraph>, <table>, <list>, <content>, <renderMultimedia>, etc. element. For example:

```
<linkHtml href="#81E32825-5BC8-46EB-8043-AE607B3819FA">Table 1</linkHtml> shows plasma clearance and half-life of gemcitabine following short infusions for typical patients by age and gender.
…
…
<table ID="81E32825-5BC8-46EB-8043-AE607B3819FA">
<caption>test table</caption>
    <thead>
        <tr .../>
    </thead>
    <tfoot>
        <tr .../>
    </tfoot>
    <tbody>
        <tr .../>
    </tbody>
</table>
```

The SPL schema also permits use of the <linkHtml> (e.g., <linkHtml name="table_1">) as a means to identify the target of a link but this is not recommended and linking by this mechanism will not be supported during rendering by the SPL standard stylesheet

4.1.8 Supplemental Patient Material

Supplemental Patient Material is a special optional section that is the last section in an SPL document. This section contains any additional material that may be included as part of the content of the package insert but not part of the major label sections, i.e., additional material past the regulatorily defined sections of labeling. Examples would be a Patient Package Insert, Medguide, or other information. Within this section, nested as subsections, can be sections identifying material as either a PPI or Medguide (LOINC codes for these are pending; see Appendix 7.6). PPI or Medguide sections should not be used unless nested within the Supplemental Patient Material section.

5. Creating the Drug Listing Data Elements Section

The drug listing data elements section of SPL contains information included in both drug listing and drug labeling[32]. Future releases of SPL may include data elements covering other areas of the content of labeling.

At the present time, only the following information should be included in the data elements section of SPL:

Active ingredient(s) - name, strength
Active moiety - name
Inactive ingredient(s) - required for parenteral products, optional for non-parenteral forms. The name of the inactive ingredient is required, strength is optional
Drug product: Proprietary name
Drug product: Nonproprietary (Established) name
Container type(s)
Quantity per container (and number of containers per container in a combination product)
National Drug Code(s) (NDC)
Drug Enforcement Administration (DEA) schedule
Dosage Form
Labeled route(s) of administration
Imprinting:[33]

- Color
- Shape
- Size
- Coating
- Scoring
- Logo (the graphic/figure that may appear on a solid preparation)
- Imprint Code (any alphanumeric information on a solid preparation)

The specific coding for each item is discussed in Table 15 below.

The following discussion addresses how to author the data elements section of the SPL. It is expected that most SPL authors will have automated tools to construct the data elements section of SPL from a template, especially if several different dosage forms or strengths of a product exist.

Information on the drug listing data elements is in 21 CRF 207 and associated guidance documents. See http://www.fda.gov/cder/drls/default.htm for detailed information on the drug listing process. Information on imprinting is in 21 CFR 206.

5.1 Conceptual View of the Model

The following is a conceptual view of the SPL drug model. (The standard diagrammatic view used by HL7 is available in the SPL specification.)

An SPL document may include information about one or more approved drug products. A drug product represented by SPL is described by the data elements in the following table:

[32] Drug listing is a process used by the FDA to maintain an inventory of drug products marketed in the United States. The regulations for the drug listing process are in Title 21 of the Code of Federal Regulations (CFR) part 207. The drug listing data elements in SPL are described in the regulation.
[33] Imprinting is described in Title 21CFR part 206. This applies to solid dosage forms though may be used for other dosage forms.

Table 12 - Conceptual view of the model for a drug product in the data elements section - single drug product with one or more package configurations

Proprietary Name	Established Name	Active Ingredients & strength	Inactive Ingredients & strength[b]	Dosage Form	Imprinting	Package Type & Quantity[a]	NDC[b,c]
		List of Active ingredients	List of inactive ingredients		List of characteristics	Package/Qty 1	NDC 1
						Package/Qty 2	NDC 2
						Package/Qty ...	NDC ...

[a] There may be one or more package/quantity/NDC triads for a proprietary name; if only one exists then there would be only one minor row in this model.
[b] Including the strength of inactive ingredients is optional unless required by regulations, e.g., in intravenous preparations.
[c] Every NDC code is unique in this model.

This table represents the simplest possible case of a drug product modeled in SPL, i.e., where there is a single drug product distributed in one or more package configurations.

A single drug product with a single strength and a single dosage form (e.g., a 75 mg tablet) would have one proprietary name and one nonproprietary (established) name. If there is only one packaging option (e.g., bottles of 100), then there would be only one NDC. If there are several packaging configurations, each configuration would have a separate NDC (e.g., bottles of 10, bottles of 100). This would result in multiple minor rows in the final 2 columns. A specific drug product/package type/quantity triad is associated with a single NDC.[34]

A slightly more complex model is where there are multiple drug products in one SPL document as illustrated in the following table.

Table 13 - Conceptual view of the model for multiple drug products in one SPL Document

Proprietary name[a]	Established Name	Active Ingredients & strength	Inactive Ingredients & strength	Dosage Form	Imprinting	Package Type & Quantity[b]	NDC[c]
Name 1		List of Active ingredients	List of inactive ingredients		List of characteristics	Package/Qty 1	NDC 1a
						Package/Qty 2	NDC 1b
						Package/Qty etc.	NDC …
Name 2		List of Active ingredients	List of inactive ingredients		List of characteristics	Package/Qty 1	NDC 2a
						Package/Qty 2	NDC 2b
						Package/Qty etc.…	NDC …
Name 3		List of Active ingredients	List of inactive ingredients		List of characteristics	Package/Qty 1	NDC 1c
						Package/Qty 2	NDC 2c
						Package/Qty etc.	NDC …

[a] Proprietary name is likely to be identical for all rows of the table, e.g., when the difference between row1 and row2 is a different dose form, but different proprietary names may exist and are accommodated in the model. Similar considerations exist for established name, i.e., it is expected that the established name will be identical for all rows in the table.
[b] There may be only one package/quantity/NDC triad for each major row in this table. The additional Package/Qty minor rows are for illustrative purposes.

[34] There should be a minor row in the table for every packaging configuration that is associated with an NDC number. This should only include packaging options that are in the How Supplied section of labeling, i.e., drug 'sample' configurations used for marketing purposes should not be included in SPL, even if associated with an NDC code.

[c] Every NDC is unique in this model.

The following is an example of a two drug products differing only by the strengths of the active ingredient:

Proprietary name	Established name	Active Ingredients	Inactive Ingredients	Dosage Form & Route	Impr.[a]	Package Type & Quantity	NDC
Gemzar	gemcitabine HCl	• gemcitabine hydrochloride[b]; 200 mg	• Manitol; 200 mg • sodium acetate; 12.5 mg	Injection, Powder, Lyophilized, For Solution; Intravenous		Vial, Single Dose; 10 ml	0002-7501-01
Gemzar	gemcitabine HCl	• gemcitabine hydrochloride; 1 g	• Manitol; 200 mg • sodium acetate; 62.5 mg	Injection, Powder, Lyophilized, For Solution; Intravenous		Vial, Single Dose; 10 ml	0002-7502-01

[a] Imprinting characteristics of the dose form (see Table 16: Imprint Codes); these are not applicable to a non-solid dose form..
[b] Unless used in the name in narrative text or titles, all names/salts should be spelled out, e.g., hydrochloride rather than HCL.

More complex examples are multicomponent products. These can take several forms, the simplest being where there are two drug products provided as components of a single drug product (even if names do not exist for the individual components, e.g., oral contraceptives). Kits, consisting of separate containers packaged together are another form of a multicomponent product. This would be approached conceptually as follows:

Table 14 - Conceptual view of the Model for Data Elements (listing elements) for a 'Multiple Component' Product

Overall Proprietary name	Overall Established Name	Proprietary Name[a]	Established Name	Active Ingred. & strength	Inactive Ingred. & strength	Dose Form	Imprinting	Comp. Pack. Type & Quan.[b]	NDC	Overall Pack. Type & Quan.	NDC[c]
		Comp 1		List of Active ingredients	List of inactive ingredients		List of characteristics				
		Comp 2		List of Active ingredients	List of inactive ingredients		List of characteristics				
		Comp 3.		List of Active ingredients	List of inactive ingredients		List of characteristics				

[a] Proprietary names may not exist for one or more of the components. Although there may be more than one package/quantity (and associated NDC number) for a component, it is unlikely in this setting.
[b] Package type is also unlikely for a component; the overall product should have this property.
[c] This is the NDC for the overall product; individual components may or may not have an NDC code.

In this example, there is an overall brand proprietary name and an overall nonproprietary (established) name. For the example of an oral contraceptive product, the package type could be a 'wheel' with a quantity of 1. However, there could be separate entries for each of the components as the minor rows of the table. For each component there may not be all the entries as there would be for a stand-alone product. For example, there may not be a proprietary name for a component, but the established name could be substituted for this value.[35]

[35] An example of coding the data elements for this situation, i.e., multiple drug products forming one drug product with a separate NDC code is not included in this version of the implementation guide..

Shown below is the structure of the SPL elements for each of the conceptual elements described above. The values for the actual elements in the SPL document that match the conceptual elements described above are in Table 15 below. (Route of administration is also included in the example though this is not necessary for the conceptual view of the model.)

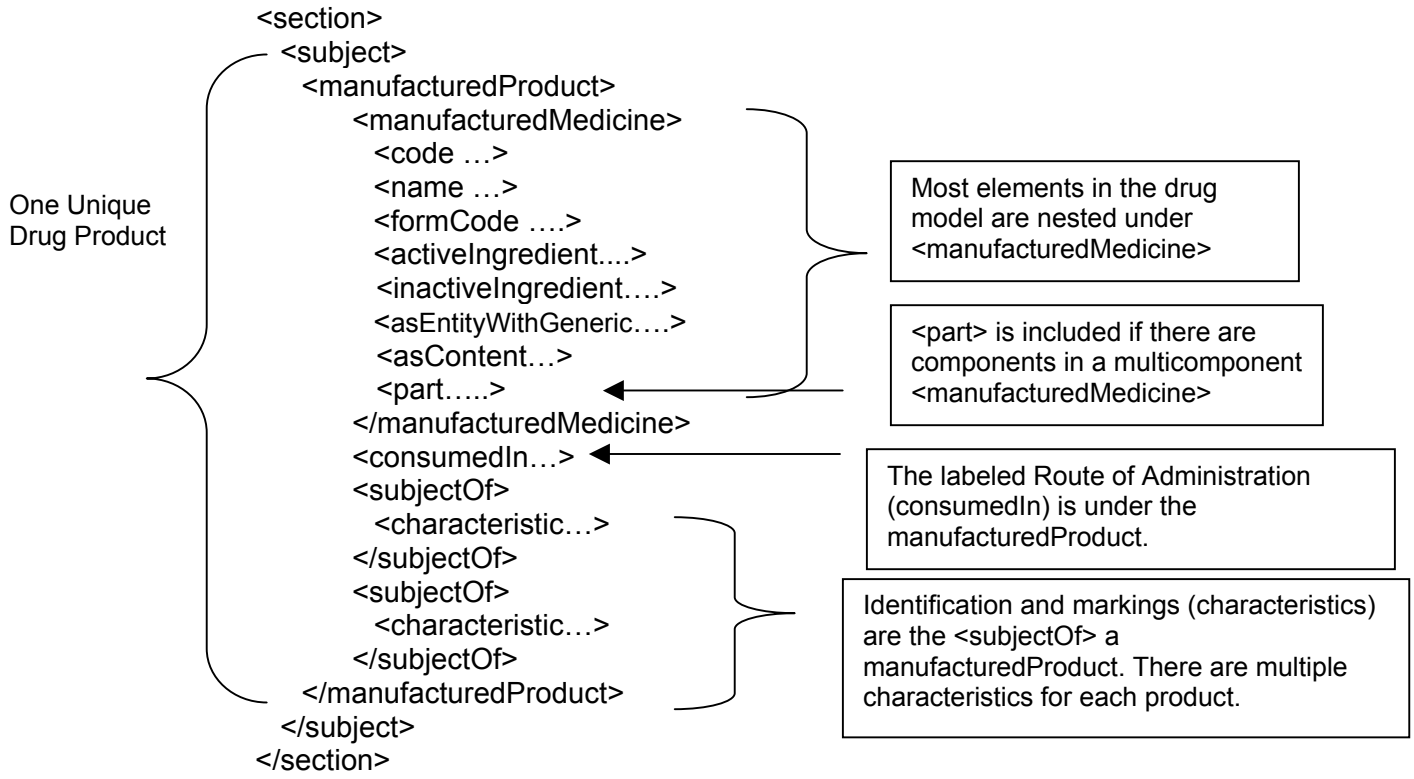


Figure 6: 'Schema' Model of SPL Data Elements section**.**

5.2 Coding the Data Elements

Table 15 below describes the actual coding of elements in an SPL document for the elements described in the conceptual view earlier. For this table, the following definitions have been used:

Active Ingredient: substance responsible for effects of a medicinal product defined by the complete molecular structure (e.g., with any counter ion) or, if the precise molecular structure is not known, by an unambiguous definition (e.g., the process used to create the substance), e.g. codeine phosphate. As discussed below, this is coded in SPL by a UNII (Unique Ingredient Identifier) provided by FDA.

Active Moiety: portion of the active ingredient responsible for the effect, if applicable (e.g., without counter ion), e.g., codeine. As discussed below, this is also coded in SPL by a UNII provided by FDA.

Proprietary Medicinal Product: includes one or more established medicinal products along with inactive ingredients and appearance and manufacturer or marketing authorization holder name, country and registration number, proprietary name, established number, marketing status, drug product type, and distributor. This is reflected by the proprietary medicinal code value under Proprietary name (the first row in the table below. An example would be Amoxil 500 mg tablet. There can be multiple packages (e.g., NDC codes) for a given proprietary medicinal package. A *Packaged Medicinal Product* is a Proprietary

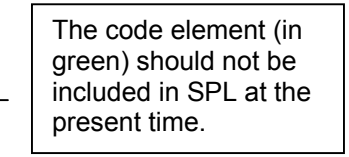
Medicinal Product in a specific package/quantity, i.e., a Packaged Medicinal Product will have an associated NDC code.[36]

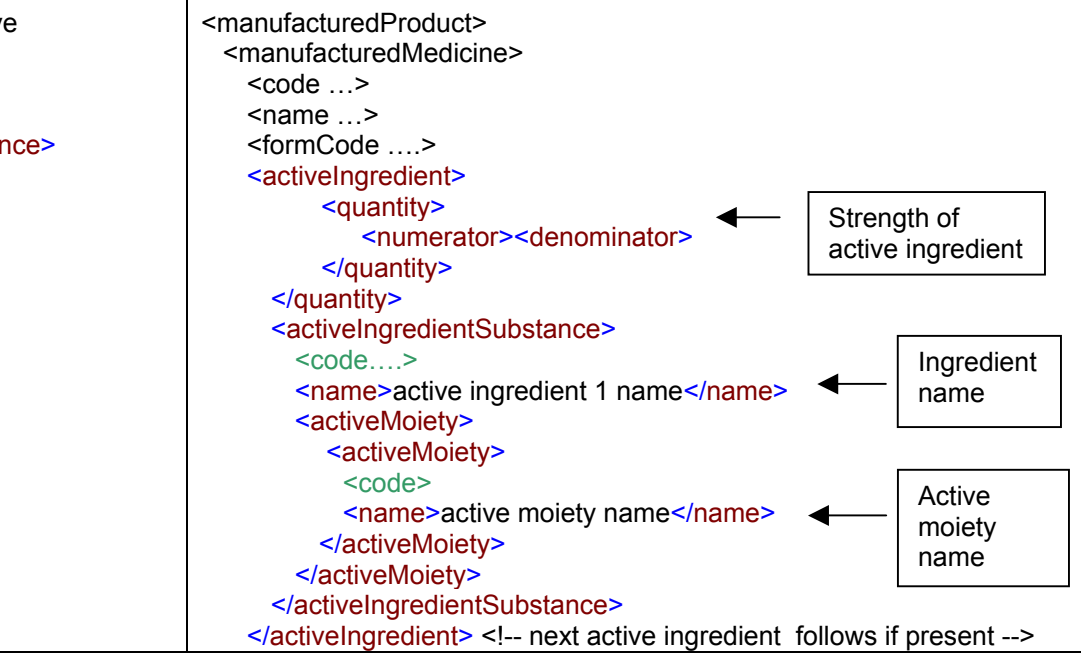
[36] Although not discussed in the table below, an Established Medicinal Product is defined as columns 2, 3 and 5 of table 1, i.e., the active ingredients and dose form but without imprinting, inactive ingredients, etc.

Table 15: Mapping and Coding of Data Elements in the Conceptual View to SPL Elements (including labeled Route of Administration)

Element Name	Schema/ FDA req.[a]	Example	Comments
Proprietary name	No/Yes	<manufacturedProduct> <manufacturedMedicine> <code code=" 0002-7501" codeSystem="2.16.840.1.113883.3.24" codeSystemName="FDA code system name"/> <name>Gemzar</name> .... </manufacturedMedicine> </manufacturedProduct>	The proprietary name is the actual text string for the proprietary name and code for the drug product is described in the drug listing regulations (21CFR207) and guidance. The code system for the <code> element under <manufacturedMedicine> is the first two components of the NDC, i.e., the labeler code and the product code. The value for the code attribute of the <code> element (i.e., <code code ="'value") is the Proprietary Medicinal Product (MP) code. This code is the product and labeler codes from the NDC, i.e., the first 4 or 5 digits of the NDC that make up the labeler code, then the 4 or 5 digits that make up the NDC product code.(e.g., 0002-7501). (Note: a Proprietary Medicinal Product represents the first 6 columns of Table 1 above, i.e., the full description of a medicinal product without the packaging information). At this juncture the codeSystem and codeSystemName are pending; therefore, this element is optional until these values are resolved. In the SPL document proprietary name is in the <name> element, a child of the <medicine>, e.g., <manufacturedProduct> <manufacturedMedicine> <code …> <name>Proprietary Medicinal Product Name here</name> <formCode ….> <activeIngredient…..> <inactiveIngredient….> <asEntityWithGeneric….> <asContent….> </manufacturedMedicine> </manufacturedProduct>

Established Name	No/Yes	<manufacturedProduct> <manufacturedMedicine> … <!-- name, form code, etc. --> <asEntityWithGeneric> <genericMedicine> <code code="to be determined" code" codeSystem="tbd " codeSystemName="tbd"/> <name>gemcitabine hydrochloride</name> </genericMedicine> </asEntityWithGeneric> … </manufacturedMedicine> </manufacturedProduct>	The actual text string for the established name for the drug as would exist in package labeling should be included in the name field under <genericMedicine>. The code system for this element has not been implemented and, therefore, the <code> element under <genericMedicine> should not be included in SPL. The established name is a child of <asEntityWithGeneric> and <genericMedicine>, e.g., <manufacturedProduct> <manufacturedMedicine> <code …> <name …> <formCode ….> <activeIngredient…..> <inactiveIngredient….> <asEntityWithGeneric> <genericMedicine> <code….> <name>established name</name> </genericMedicine> </asEntityWithGeneric> <asContent….> </manufacturedMedicine> <manufacturedProduct> All names/salts should be spelled out, e.g., 'gemcitabine hydrochloride' rather than 'gemcitabine HCL' unless when used in the name in narrative text or titles,



Active Ingredients (Active Moieties)	No/Yes	<manufacturedProduct> <manufacturedMedicine> <!-- name, form code, etc. --> … <activeIngredient> <quantity> <numerator value="500" unit="MG"/> <denominator value="1"/> </quantity> <activeIngredientSubstance> <code code="SRS determined code" codeSystem="FDA SRS system" codeSystemName="SRS code name"/> <name>ingredient 1</name> <activeMoiety> <activeMoiety> <code code="SRS determined code" codeSystem="FDA SRS system" codeSystemName="SRS code name"/> <name>ingredient 1 active moiety</name> </activeMoiety> </activeMoiety> </activeIngredientSubstance> </activeIngredient> <activeIngredient> …..	The active ingredient name is part of a complex structure. <activeIngredient> is a child of <manufacturedMedicine>. <activeIngredient> has several children. The <quantity> is a direct descendent of the <activeIngredient>, and represents the strength of the active ingredient. The value for units should be taken from the table at http://www.fda.gov/cder/dsm/DRG/drg00501.htm. The <activeIngredientSubstance> child contains the name of the activeIngredient and SRS code for the ingredient; it also contains the <activeMoiety><activeMoiety> children, the latter of which has its own code and name. (Note that <activeMoiety> is repeated twice.) Active ingredient, active moiety, activeMoietyEntity, and inactive ingredient are defined in the SPL specification. The code system for this element and all nested elements will be the FDA Substance Registration System (SRS), which is under development; until SRS is completed, the <code> element should not be used for activeIngredient or activeMoiety. If either element is included in SPL, this information will not be processed by FDA. The value for the codes will be the UNII (unique ingredient identifier). 

			<inactiveIngredient….> <!-- inactive ingredients follow… --> <asEntityWithGeneric….> <asContent….> </medicine> </manufacturedProduct> Strength is indicated by the <quantity> tag, with a <numerator> and <denominator>. The numerator and denominator each have 'value' and 'unit' attributes. The unit attribute is optional as in the example to the left. Commonly used quantity units are: MG, UGM, GM, ML, and UL. Units must be from the table at http://www.fda.gov/cder/dsm/DRG/drg00501.htm For a solution, an example could be <numerator value = "30" unit="MG"><denominator value="5" unit="ML"> At this time it is anticipated that FDA will assign the UNII codes for ingredients after the submission of an original SPL (i.e., the first version of a document). SPL should be submitted with only the ingredient name fields included. All subsequent versions of a specific SPL document must the contain the coded UNII values for each ingredient that were assigned by FDA with the original submission (unless ingredients have changed). At this time FDA will insert UNII codes for the original submission when the document is exported from FDA, i.e., the sponsor will not be responsible for inserting UNII codes during the development of the initial product SPL. This may be altered in the future.

Inactive ingredients	No/Yes (required for parenteral preparations only)	<manufacturedProduct> <manufacturedMedicine> <!-- name, form code, etc. --> <activeIngredient ….> </activeIngredient> <inactiveIngredient> <quantity> <numerator value="200" unit="MG"/> <denominator value="1"/> </quantity> <inactiveIngredientSubstance> <code code="SRS determined code" codeSystem="FDA SRS system" codeSystemName="SRS code name"/> <name>mannitol</name> </inactiveIngredientSubstance> </inactiveIngredient> ……	The Inactive Ingredient element description parallels the Active Ingredient element. However, an 'active moiety' child is not applicable for an inactiveIngredient. In addition, quantity is optional for inactive ingredients. The code system for this element and all nested elements in inactive ingredients will be the FDA SRS system; until SRS is completed, the <code> element should not be used, if used, this information will not be processed by FDA <inactiveIngredient> <quantity> <numerator/><denominator/> </quantity> <inactiveIngredientSubstance> <code….> <name>inactive ingredient name </name> </inactiveIngredientSubstance> </inactiveIngredient> <!-- next inactive ingredient …..> Strength is optional for inactive ingredients in non-intravenous products Inactive ingredients are optional for nonparenteral products; for both parenteral and nonparenteral products, strength is optional. At the present time listing of inactive ingredients should parallel what is required for printed labeling and SPL should include the inactive ingredient currently specified in printed labeling. In this context, most (but not every) inactive ingredients for non-oral dose forms are required in printed labeling (and thus SPL) but for oral dose forms inactive ingredients are optional. However, if an inactive ingredient is listed in the present content of labeling, it should be included in the data elements as well.

Dosage Form	No/Yes	<manufacturedProduct> <manufacturedMedicine> <name…> <formCode code="716" codeSystem="2.16.840.1.113883.3.24.1.1" codeSystemName="FDA DRG 201" displayName="INJECTION, LIPID COMPLEX" /> <activeIngredient....> <inactiveIngredient….> <asEntityWithGeneric….> <asContent….> </medicine> </manufacturedProduct>	Dosage form is coded in the 'formCode' element, a direct child of the medicine element. At the present time the value for the codeSystem attribute is pending. The value for the code and displayName attributes should be the appropriate value from the table at http://www.fda.gov/cder/dsm/drg/drg00201.htm. The general form of this element is: <manufacturedProduct> <manufacturedMedicine> <code …> <name …> <formCode code="value from DRG 201" codeSystemName="FDA OID for Table DRG 201" codeSystemName="Name for Table DRG 201" displayName="value from DRG 201"/> <activeIngredient....> <inactiveIngredient….> <asEntityWithGeneric….> <asContent….> </manufacturedMedicine> </manufacturedProduct>

Labeled Route of Administration	No/Yes	<manufacturedProduct> …….. <consumedIn> <substanceAdministration> <routeCode code="valyue from NCI thesaurus 313" codeSystem="NCI Thesaurus" codeSystemName="2.16.840.1.113883.3.26.1.1" displayName="ENTERAL"/> </substanceAdministration> </consumedIn> </manufacturedProduct>	Labeled route of administration (see definition in the SPL specification) is coded in the routeCode element, a child of the <manufacturedProduct>, <consumedIn>, and <substanceAdministration> elements. The codeSystem to be used for Route of Administration will be the NCI thesaurus; the OID for this thesaurus is 2.16.840.1.113883.3.26.1.1. However, this has not been implemented in the NCI thesaurus at the present time and the value for code and codeSystemName should not be included with SPL (See Section 7.8 for additional information regarding the NCI Thesaurus.) The values for this element from the NCI Thesaurus will correspond to entries in the table at http://www.fda.gov/cder/dsm/DRG/drg00301.htm. However, the exact codes will be the unique codes from the NCI thesaurus. For example: <manufacturedProduct> …….. <consumedIn> <substanceAdministration> <routeCode code="value from NCI thesaurus" codeSystem="2.16.840.1.113883.3.26.1.1" codeSystemName="Name for NCI thesarus" displayName="ENTERAL”> </substanceAdministration> </consumedIn> </manufacturedProduct> At the present time the only values are the display name (e.g., “ENTERAL” for the displayName from table C-DRG-00301). Therefore, until use of the NCI thesaurus is implemented for SPL, the element in this example would be coded as: <routeCode displayName="ENTERAL”> If more than one route of administration is approved, <consumedIn> blocks are repeated, e.g., <manufacturedProduct> …….. <consumedIn> <!--- first route or administration --> …… </consumedIn> <consumedIn> <!--- second route of administration--> </consumedIn> </manufacturedProduct>

Imprinting	No/Yes (for solid dosage forms)		Imprinting codes are the characteristics of a drug product (e.g., shape, color, logo, etc.) that are included as <characteristic> of the <manufacturedProduct> in the data elements section of SPL. These are described separately in Table 16: Imprint Codes. (Note: this is different from 'Imprint', a specific Imprinting code described in Table 16: Imprint Codes).

Package Type and Quantity	No/Yes	<manufacturedProduct> <manufacturedMedicine> …… <asContent> <quantity> <quantity> <numerator value="100" unit="MG"/> <denominator value="1"/> </quantity> <containerPackagedMedicine> <formCode code="VIAL" codeSystem="2.16.840.1.113883.3.26.1.1" codeSystemName="NCI Thesaurus" displayName="VIAL" /> <code code="0002-7502-01" codeSystem="2.16.840.1.113883.6.69" codeSystemName="NDC"/> </containerPackagedMedicine > </asContent> </manufacturedMedicine> </manufacturedProduct>	The package type is contained in the <manufacturedMedicine> as the child of <asContent><containerPackagedMedicine> Quantity is the direct child of <asContent> and has the same structure as <quantity> when used with other elements (e.g., active ingredients). Package Type is the value of <formCode> under <containerPackagedMedicine>. Similar to labeled Route of Administration (see above), the codeSystem to be used for Package Type/formCode will be the NCI thesaurus; the OID for this thesaurus is 2.16.840.1.113883.3.26.1.1. However, this has not been implemented in the NCI thesaurus at the present time and the value for code and codeSystemName should not be included with SPL The values for this element from the NCI Thesuarus will correspond to entries in the table at http://www.fda.gov/cder/dsm/DRG/drg00907.htm. However, the exact codes will be the unique codes from the NCI thesaurus. For example: <manufacturedProduct> <manufacturedMedicine> <asContent> <quantity> <numerator value="numerator value" unit="units value"/> <denominator value="denominator value"/> </quantity> <containerPackagedMedicine> <formCode code="VIAL" codeSystem="2.16.840.1.113883.3.26.1.1" codeSystemName="NCI Thesaurus" displayName="VIAL" /> <code code="0002-7502-01" codeSystem="2.16.840.1.113883.6.69" codeSystemName="NDC"/> </containerPackagedMedicine> </asContent> </manufacturedMedicine > </manufacturedProduct> At the present time the only values are the display name (e.g., “VIAL” for the displayName from table C-DRG-00907). Therefore, until use of the NCI thesaurus is implemented for SPL, the element in this example would be coded as: <formCode displayName="VIAL”>

DEA number	No/Required only for controlled substance (see text)	<manufacturedProduct> <manufacturedMedicine> …… </manufacturedMedicine> <subjectOf> <policy classCode=”DEADrugSchedule”> <code code="II" codeSystem="OID for DEA schedule"> </policy> </subjectOf> </manufacturedProduct>	DEA number is included in the SPL model as the <code> for <policy>, a <subjectOf> a <manufacturedProduct>. The structure is: <manufacturedProduct> <manufacturedMedicine> …… </manufacturedMedicine> <subjectOf> <policy classCode=”DEADrugSchedule”> <code code="value for DEA schedule" codeSystem="OID for DEA schedule" codeSystemName="DEA schedule" displayName=""/> </policy> </subjectOf> </manufacturedProduct> This field is required if an SPL contains a product that is a DEA scheduled substance. The code system for DEA schedule is pending at this time. Because of this, the coded schedule value (I,II, III, IV, or V) should be used in an <originalText> child of <code>, i.e., <policy classCode=”DEADrugSchedule”> <code> <originalText>III</originalText> </code> </policy>

NDC Number	No/Yes	See Package Type and quantity above.	The NDC number is the code child of the <containerPackagedMedicine> element; as such it is directly associated with a container. In SPL release 2, a <container> has a <quantity> and a <containerPackagedMedicine>. The latter has a <formCode> (type of package/container) and associated NDC code <code). The structure is as follows: <manufacturedProduct> <manufacturedMedicine> …… <asContent> <quantity> <numerator value="numerator value" unit="units value"/><denominator value="denominator value"/> </quantity> <containerPackagedMedicine> <formCode displayName="VIAL”> <code code="0002-7502-01" codeSystem="2.16.840.1.113883.6.69" codeSystemName="NDC"/> </containerPackagedMedicine> </asContent> </manufacturedMedicine> </manufacturedProduct> The codeSystemfor the <formCode> is always "2.16.840.1.113883.6.69" (LOINC) and the codeSystemName is always "NDC" . The value for the NDC number is the code itself.
Approval information			<approval> is included in the SPL schema in anticipation of the physicians labeling rule. It is a child of <subjectOf> from < manufactured product> and includes the information FDA NDA number, date of first approval <author><time> and labeling status (e.g., active, pending, withdrawn). <subjectOf><approval> should not be used in SPL at this time. Guidelines on the use of this element will be discussed in a future version of this guide.

[a] Whether the element is required by the schema and/or FDA.

Imprinting and other descriptive information for solid dosage forms is captured by means of a <characteristic> that are the <subjectOf> a <manufacturedProduct>. The values for the <code> element child of <characteristic> are the codes that describe the physical markings and identification of the product, if available. If no code is available, the <text> field may be used.

For a characteristic (e.g., shape), where no code system exists (currently all characteristics; no code systems have been adopted at this time) :

```
<section>
  <subject>
    <manufacturedProduct>
      <manufacturedMedicine>
      …..
      </manufacturedMedicine>
      <subjectOf>
        <characteristic classCode="OBS">
            <code code="imprint code, e.g., FDASHAPE"/>
            <text>round</text>
        </characteristic>
      </subjectOf>
    </manufacturedProduct>
  </subject>
</section>
```

.
If a code system did exist, the following form would be used:

```
<characteristic classCode="OBS">
    <code code="FDA Characteristic"/>
    <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CE">
    <xsi:code="code for specific value" xsi:codeSystem="OID for code system">
    </value>
</characteristic>
```

The required characteristics include:

- Color
- Shape
- Size
- Coating
- Scoring
- Logo
- Imprint Code

Each of these has a corresponding HL7 code, i.e.,

- FDACOLOR
- FDASHAPE
- FDASIZE
- FDACOATING
- FDASCORING
- FDAIMPRINTCD
- FDALOGO

It is expected that each characteristic will be present for every solid dosage form. If a characteristic is not applicable (e.g., shape of a suspension), then it can be omitted

Since the values for some of these characteristics are not standardized as yet, authors should use discretion in describing the characteristics with terms that communicate readily, e.g., round is preferred to circular, oval to ovoid or elliptical, etc.

Table 16: Imprint Codes

Color	FDACOLOR	There is no code system for color at present. Values should be entered as follows: <subjectOf> <characteristic classCode="OBS"> <id root="78AF9883-92D3-4028-8AFB-B00BC8220117"/> <code code="FDACOLOR"/> <text>white</text> </characteristic> </subjectOf> The use of this characteristic is to generally identify the color of the solid dosage form; it is expected that the value of text for this element will be the same as the description used in the content of labeling.
Scoring	FDASCORING	There is no code system for scoring at present. However, the following values should be used in the text field: 0 – non-solid dose forms, powders, etc. 1 - a solid dosage form that is not scored. 2 or greater - The maximum number of possible equal pieces that a dosage form could result in based on the scoring. <subjectOf> <characteristic classCode="OBS"> <id root="78AF9883-92D3-4028-8AFB-B00BC8220117"/> <code code="FDASCORING"/> <text>2</text> </characteristic> </subjectOf>
Shape	FDASHAPE	There is no code system for shape at present. Values should be entered as follows pending a coding system: <subjectOf> <characteristic classCode="OBS"> <id root="23AF9883-92D3-4028-8AFB-B00BC8220117"/> <code code="FDASHAPE"/> <text>oval</text> </characteristic> </subjectOf>
Size	FDASIZE	There is no code system for size at present and a coding system is unlikely for this element. Until a final nomenclature for this element is resolved, the longest side of a solid dosage form in millimeters should be recorded.as a <text> element, e.g., : <subjectOf>

		<characteristic classCode="OBS"> <id root="23AF9883-92D3-4028-8AFB-B00BC8220117"/> <code code="FDASIZE"/> <text>55</text> </characteristic> </subjectOf>
Coating	FDACOATING	There is no code system for coating at present. Values should be entered as "Yes" or "No" in the <text> element depending on whether a coating is present . See the example for shape above.
Logo	FDALOGO	There is no code system for logo and a codesystem for this element is unlikely. Values should be entered as <text> elements if possible. See the examples above. The value for logo should be a text description of the logo. In a future revision the logo observation may refer to an image file
Imprint	FDAIMPRINTCD	There is no code system for imprint nor is a code system likely. Values should be entered as <text> elements as in the examples above. The value for Imprint should be the actual text that appears on a solid dosage form. Multiple lines should be separated by a semi-colon, e.g., <subjectOf> <characteristic classCode="OBS"> <id root="89AF9883-92D3-4028-8AFB-B00BC8220117"/> <code code="FDAIMPRINTCD"/> <text>Line 1 of imprint;Line 2 of imprint</text> </characteristic> </subjectOf>

All data elements for one unique product (corresponding to a major row in the conceptual view table) are contained within a single <component><section> in the SPL body; each unique product (<manufacturedProduct>) will have its own section. This means that machine processable information about a drug product will be aggregated in one section, even though the textual (human readable) information may tend to be scattered in different sections in the narrative text (e.g., in the Description and How Supplied sections). Sections that contain these structured data elements that are built around <manufacturedProduct> will only contain structured data and not text.

An example of the XML markup of machine processable structured data elements that describe a drug product (one section per product) is shown below. By convention it is requested that the <manufacturedProduct> sections be present as the initial sections in the body following the header elements.

```
<Header Section….>
……
<Body Section …..>
<component>
    <structuredBody>
        <component><!-- this component/section contains all the data elements -->
            <section>
                <id root="81E32825-5BC8-46EB-8043-AE607B3819FA " />
                <!-- Each proprietary drug product as a separate subject -->
                <subject>
                    <manufacturedProduct>
                        <manufacturedMedicine>
                            ….
                        </manufacturedMedicine>
                        <consumedIn>
                            ….
```

GUID for section

Drug information (see above) for drug product 1

Labeled Route of Administration for drug product 1

```
                </consumedIn>
                    <subjectOf>
                        <characteristic>
                        .....
                        </characteristic>
                    </subjectOf>
                </manufacturedProduct>
            </subject>
            <subject>
                ..............
                ..............
            </subject>

            <!-- - - - - - - -          end of  drug products/data elements    - - - - - - - - -->

    <!-- - - - - - - -   Now start Narrative section - - - - -->
    <!-- - - - - - - -   e.g., Description section  - - - - - - - - - -->
        <component> <!-- This component contains all the content of labeling -->
            <section>
                <id root="887C4DF62-7F0B-4267-A1A7-50F22BA35446"/>
                <code code="34089-3" codeSystem="2.16.840.1.113883.6.1"
                codeSystemName="LOINC" displayName="Description"/>
                <title>DESCRIPTION</title>
        ……
        ……..
```

Identification and Markings (Imprinting) for drug product 1

Start of Drug Product 2

Figure 7: Model of SPL 'Drug Elements' section.

The following example shows a drug product section for an SPL document with insertion of more detailed data elements. Note that this identical format would be repeated for each row in the conceptual view table above.

```
<component>
    <section>
        <id root="675D3B70-0457-4F92-AA4A-8B30928A32A9"/>
        <subject>
            <manufacturedProduct>
                <manufacturedMedicine>
                    <code code="0004-2601" codeSystem="FDA code system"
                    codeSystemName="FDA code name"/>
                    <name>ExampleDrug</name>
                    <formCode displayName="TAB COATED"/>
                    <activeIngredient>
                        <quantity>
                            <numerator value="87.5" unit="MG"/>
                            <denominator value="1"/>
                        </quantity>
```

Product Code (See Proprietary Name in Table 15

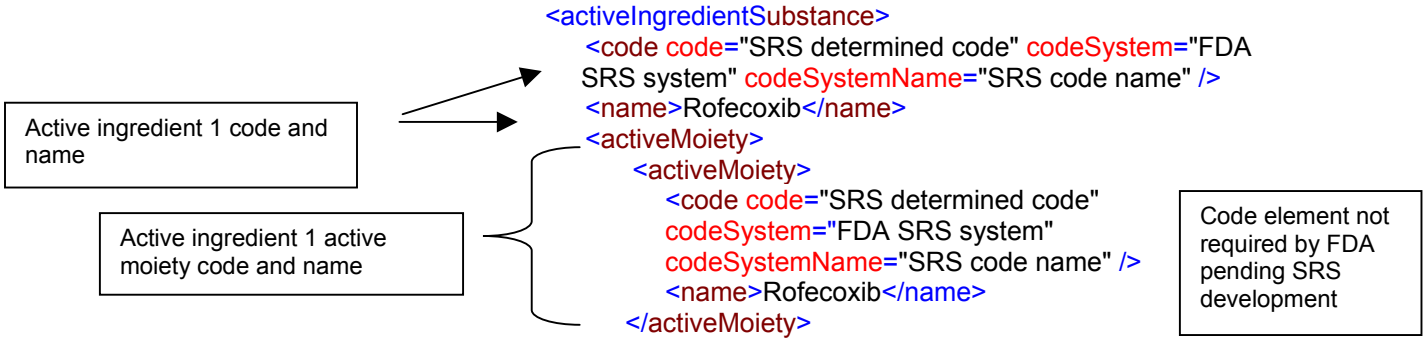
Proprietary Name

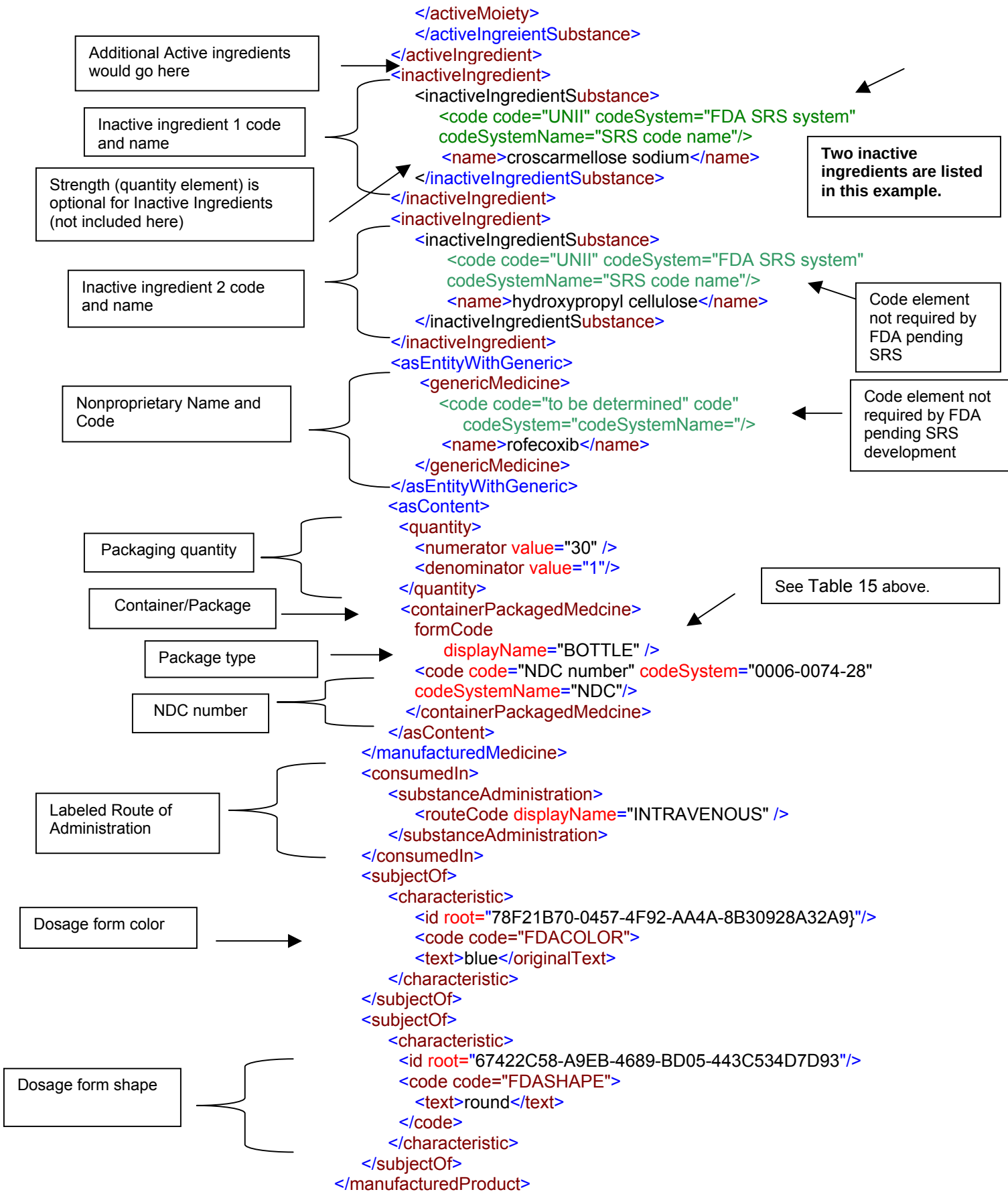
Dosage Form Name

Active Ingredient 1 strength

See Table 15 above.

See Table 15 above.


<activeIngredientSubstance>
<code code="SRS determined code" codeSystem="FDA SRS system" codeSystemName="SRS code name" />
<name>Rofecoxib</name>
<activeMoiety>
<activeMoiety>
<code code="SRS determined code"
codeSystem="FDA SRS system"
codeSystemName="SRS code name" />
<name>Rofecoxib</name>
</activeMoiety>
Active ingredient 1 code and name
Active ingredient 1 active moiety code and name
Code element not required by FDA pending SRS development


Additional Active ingredients would go here
Inactive ingredient 1 code and name
Strength (quantity element) is optional for Inactive Ingredients (not included here)
Inactive ingredient 2 code and name
Nonproprietary Name and Code
Packaging quantity
Container/Package
Package type
NDC number
Labeled Route of Administration
Dosage form color
Dosage form shape
Two inactive ingredients are listed in this example.
Code element not required by FDA pending SRS
Code element not required by FDA pending SRS development
See Table 15 above.
</activeMoiety>
</activeIngreientSubstance>
</activeIngredient>
<inactiveIngredient>
<inactiveIngredientSubstance>
<code code="UNII" codeSystem="FDA SRS system"
codeSystemName="SRS code name"/>
<name>croscarmellose sodium</name>
</inactiveIngredientSubstance>
</inactiveIngredient>
<inactiveIngredient>
<inactiveIngredientSubstance>
<code code="UNII" codeSystem="FDA SRS system"
codeSystemName="SRS code name"/>
<name>hydroxypropyl cellulose</name>
</inactiveIngredientSubstance>
</inactiveIngredient>
<asEntityWithGeneric>
<genericMedicine>
<code code="to be determined" code"
codeSystem="codeSystemName="/>
<name>rofecoxib</name>
</genericMedicine>
</asEntityWithGeneric>
<asContent>
<quantity>
<numerator value="30" />
<denominator value="1"/>
</quantity>
<containerPackagedMedcine>
formCode
displayName="BOTTLE" />
<code code="NDC number" codeSystem="0006-0074-28"
codeSystemName="NDC"/>
</containerPackagedMedcine>
</asContent>
</manufacturedMedicine>
<consumedIn>
<substanceAdministration>
<routeCode displayName="INTRAVENOUS" />
</substanceAdministration>
</consumedIn>
<subjectOf>
<characteristic>
<id root="78F21B70-0457-4F92-AA4A-8B30928A32A9}"/>
<code code="FDACOLOR">
<text>blue</originalText>
</characteristic>
</subjectOf>
<subjectOf>
<characteristic>
<id root="67422C58-A9EB-4689-BD05-443C534D7D93"/>
<code code="FDASHAPE">
<text>round</text>
</code>
</characteristic>
</subjectOf>
</manufacturedProduct>

```
            </subject>
            <component> <!-- This component contains all the content of labeling -->
                <section>
                    <id root="887C4DF62-7F0B-4267-A1A7-50F22BA35446"/>
                        <code code="34089-3" codeSystem="2.16.840.1.113883.6.1"
                    codeSystemName="LOINC" displayName="Description"/>
                    <title>DESCRIPTION</title>


                    ..........
                    ..........
                </section>
            </component>
        </section>
    </component>

    ........
```

Figure 8: Annotated Example of SPL 'Drug Product' section.

The following is an fictitious example of the simplest form of a multicomponent product, i.e., where a single package (with one NDC number) combines two medications. (In this case a 'Contraceptive medicine dial pack' containing 20 'estrogens' and 10 'progestins'.) Neither of the <part>s are in a separate container (e.g., a kit), nor is there a DEA schedule for the drug. Characteristics of the 'overall' product are not coded in this example. More complex examples, i.e., where there are subcontainers and NDC or DEA numbers on the individual parts are straightforward in the SPL model. (Note that the form codes are not accurate in this example.)

```
…..
<structuredBody>
  <component>
    <section>
        <subject>
            <manufacturedProduct>
                <manufacturedMedicine>
                    <name>Contraceptive Example Medication</name>
                    <asContent>
                        <quantity>
                            <numerator value="1" />
                                <denominator value="1"/>
                        </quantity>
                          <containerPackagedMedciine>
                            <code code="0002-7502-01" codeSystem="2.16.840.1.113883.6.69"
                                codeSystemName="NDC"/>
                                 <formCode code="NCI thesaurus code"
                                codeSystem="2.16.840.1.113883.3.26.1.1"
                                codeSystemName="NCI thesaurus"   displayName="DIAL PACK">
                        </containedPackagedMedicine>
                    </asContent>
                    <part>
                        <quantity>
                            <numerator value="20" />
                                <denominator value="1"/>
                        </quantity>
                        <partMedicine>
                            <name>Estrogen Component of Contraceptive Medication</name>
                            <formCode code="500" codeSystem="2.16.840.1.113883.3.24.1.1"
                                codeSystemName="FDA DRG 201" displayName="TABLET" />
                            <activeIngredient>
                                <quantity>
                                    <numerator value=".3" units=”MG” />
                                    <denominator value="1" units=”MG” />
                                </quantity>
                                <activeIngredientSubstance>
                                    <code code="SRS determined code" codeSystem="FDA SRS system"
                                        codeSystemName="SRS code name"/>
                                    <name>estrogen</name>
                                    <activeMoiety>
                                        <activeMoiety>
                                            <code code="SRS determined code"
                                                codeSystem="FDA SRS system"
                                                codeSystemName="SRS code name"/>
                                            <name>estradiol</name>
                                        </activeMoiety>
                                    </activeMoiety>
                                <activeIngredientSubstance>
                            </activeIngredient>
                            <inactiveIngredient>
                                <inactiveIngredientSubstance>
                                    <code code="SRS determined code" codeSystem="FDA SRS system"
                                        codeSystemName="SRS code name"/>
                                    <name>talc</name>
```

```
                    <inactiveIngredientSubstance>
                </inactiveIngredient>
                <inactiveIngredient>
                    <inactiveIngredientSubstance>
                        <code code="SRS determined code" codeSystem="FDA SRS system"
                            codeSystemName="SRS code name"/>
                        <name>Yellow Dye #8</name>
                    <inactiveIngredientSubstance>
                </inactiveIngredient>
            </partMedicine>
            <subjectOf>
                <characteristic classCode="OBS">
                    <code code="FDASHAPE"/>
                    <text>round</text>
                </characteristic>
            </subjectOf>
            <subjectOf>
                <characteristic classCode="OBS">
                    <code code="FDASCORING"/>
                    <text>2</text>
                </characteristic>
            </subjectOf>
            <subjectOf>
                <characteristic classCode="OBS">
                    <code code="FDACOLOR"/>
                    <text>white</text>
                </characteristic>
            </subjectOf>
            <consumedIn>
                <substanceAdministration>
                    <routeCode code="313" codeSystem="2.16.840.1.113883.3.24.1.2"
                        codeSystemName="FDA DRG 301" displayName="ENTERAL"/>
                </substanceAdministration>
            </consumedIn>
        </part>
        <part>
            <quantity>
                <numerator value="10" />
                    <denominator value="1"/>
            </quantity>
            <partMedicine>
                <name>Progesterone Component of Contraceptive Medication</name>
                <formCode code="500" codeSystem="2.16.840.1.113883.3.24.1.1"
                    codeSystemName="FDA DRG 201" displayName="TABLET" />
                <activeIngredient>
                    <quantity>
                        <numerator value=".25" units=”MG” />
                        <denominator value="1" units=”MG” />
                    </quantity>
                    <activeIngredientSubstance>
                        <code code="SRS determined code" codeSystem="FDA SRS system"
                            codeSystemName="SRS code name"/>
                        <name>estrogen</name>
```

```
                                    <activeMoiety>
                                        <activeMoiety>
                                            <code code="SRS determined code"
                                                codeSystem="FDA SRS system"
                                                codeSystemName="SRS code name"/>
                                            <name>estradiol</name>
                                        </activeMoiety>
                                    </activeMoiety>
                                <activeIngredientSubstance>
                            </activeIngredient>
                            <inactiveIngredient>
                                <inactiveIngredientSubstance>
                                    <code code="SRS determined code" codeSystem="FDA SRS system"
                                        codeSystemName="SRS code name"/>
                                    <name>talc</name>
                                <inactiveIngredientSubstance>
                            </inactiveIngredient>
                        </partMedicine>
                        <subjectOf>
                            <characteristic classCode="OBS">
                                <code code="FDASHAPE"/>
                                <text>square</text>
                            </characteristic>
                        </subjectOf>
                        <subjectOf>
                            <characteristic classCode="OBS">
                                <code code="FDASCORING"/>
                                <text>1</text>
                            </characteristic>
                        </subjectOf>
                        <subjectOf>
                            <characteristic classCode="OBS">
                                <code code="FDACOLOR"/>
                                <text>blue</text>
                            </characteristic>
                        </subjectOf>
                        <consumedIn>
                            <substanceAdministration>
                                <routeCode code="313" codeSystem="2.16.840.1.113883.3.24.1.2"
                                    codeSystemName="FDA DRG 301" displayName="ENTERAL"/>
                            </substanceAdministration>
                        </consumedIn>
                    </part>
                </manufacturedMedicine>
            </subjectOf>
        </manufacturedProduct>
    </subject>
</section>
</component>
<component>
    <section>
        <text>……  <!—content of labeling…-->
        ……..
```

Figure 9: Example of SPL 'Drug Product' section for combination (multiple component) drug product.

6. Submitting SPL to FDA

See FDA guidance to industry: *Regulatory Submissions in Electronic Format – Content of Labeling* for information on submitting SPL to FDA. This guidance may be found on the FDA web site at http://www.fda.gov/cder/guidance/guidance.htm .

7. Appendices

7.1 Benefits of an XML approach to SPL

The benefits that accrue to all stakeholders in the drug regulatory process by adoption of Structured Product Labeling (SPL) make the business case for implementation of SPL overwhelming. The benefits of SPL derive from use of standard, universally adopted information standards such as XML, from the specific aspects of the SPL model for describing prescription drug content, and from adoption of an open standard for SPL.

Specific benefits of SPL include:

- Stakeholders can develop automated, customized presentations of SPL content to reduce medical errors and improve patient care.
- SPL can be used by decision-support systems to improve patient care and reduce medical errors.
- Communication within industry (e.g., business-business communication) is facilitated by the ability to use SPL with XML-compliant tools or services.
- The use of well-defined vocabularies and coding systems within SPL enables uniform and unambiguous description of prescription drug products for data information systems
- XML-based transformations of SPL content by third-parties can increase the information and medical value of SPL documents while ensuring the integrity of FDA-approved labeling content.
- Internal business processes can repurpose the data contained within SPL through the use of XML-compliant databases.
- SPL insures data integrity of labeling content (and other SPL data elements) between industry and FDA databases.
- XML compliant consumer and health practitioner tools can use SPL in multiple settings.
- SPL can be readily integrated with HL7-based hospital information systems due to compliance with CDA architecture.
- Tools or objects that implement the standard can be utilized across all instances of SPL.
- Labeling content in SPL is not tied to proprietary tooling, allowing the development of SPL documents by different tools while retaining compatibility.
- SPL-associated XML stylesheets allow consistent presentation (rendering) of label content across different package inserts.
- Labeling changes and updates can be transmitted and immediately integrated into information systems.
- Non-rendered data elements (e.g., metadata about the SPL document or tagged content abstracted from the content of labeling) can be encompassed by the SPL standard to allow SPL data to integrate with other FDA and stakeholder systems.

These benefits represent only a fraction of the extensive benefits that accrue from adoption of SPL. The principles underlying these benefits are discussed below.

7.1.1 SPL as an Open Standard:

SPL has been adopted as an open standard by Health Level Seven (HL7), an ANSI-accredited Standards Developing Organization (SDO). SPL development within HL7 has followed a well-defined, rigorous, ANSI-specified set of standards development procedures that has ensured consensus, openness and balance of interest among all participants in the development process. Participants in this process have included representatives of the pharmaceutical industry, regulatory participants from the United States, and observers from other international agencies, domain and technical experts from HL7, health care providers, and software vendors.

The SPL document standard is defined by an XML schema; XML schemas are also an open standard supported by the World Wide Web consortium (W3C).[37] The vast penetration of XML technology in business and consumer products allows SPL to leverage the large number of tools available that support XML and

[37] See http://www.w3c.org/XML/.

other W3C standards[38]. The use of the HL7 information model and HL7 modeling conventions for the development of SPL ensures that SPL documents (also referred to as instances) are universally readable and unambiguously understood to all users of SPL documents and the developers of SPL tooling.[39] .

As an open standard, SPL facilitates interoperability between systems. This provides substantial benefits to users through the reduced cost of software applications due to marketplace competition and the decreased need to customize applications around a proprietary standard. As an open standard, developers benefit substantially by full, immediate access to the information necessary to implement SPL systems. As noted earlier, adherence to a standard dramatically improves integration different software applications and permits increased sophistication of software systems based on SPL.

The HL7 ANSI-mandated open development process ensures that all stakeholder concerns are addressed during the process of future enhancements to the standard. The publicly-open HL7 consensus process allows the contribution from a far greater array of participants than would be accessible to FDA in a more restricted standards-development approach. This permits a substantial number of individuals to contribute to the standard who otherwise would be excluded.

Overall, the use of an open standard such as SPL enables direct and predictable industry-to-FDA communication of package insert information, with subsequent direct communication of content to health care professionals and other third parties in a manner previously not possible with earlier FDA standards, e.g., PDF.

7.1.2 Machine Processability & Data Integration

As an XML standard SPL contains machine-readable context (i.e., tagged) information. Using XML enables the textual content to also be represented as data, i.e., "entries," in a data layer not visible to the human reader in the document but machine readable.

By structurally and semantically identifying content in this fashion, standardized data exchange between systems can occur. It permits business-to-business, industry-to-regulatory agency, and regulatory agency-to-public data transfer in a seamless manner. Knowledge of the sending or receiving system is not needed as long as there is conformance to the standard; in addition, since the definition of SPL (i.e., the XML schema for SPL) is an open standard, others systems can readily integrate structured information from SPL documents. During the exchange across systems, the need for additional transformation steps is eliminated.

Through use of XML, SPL can contain machine processable information separate from labeling content, i.e., 'data elements' related to or derived from label content but not strictly present in an identical format in current label content. For example, a tagged list of inactive ingredients may not appear as such in the content of a package insert but could be included as machine-processable data elements included by the SPL author. Data elements (or any tagged information) can serve as data for other systems separate from SPL, e.g., drug registration systems.[40]

The structure and machine readability of SPL also enables certain previously manual review functions within FDA and industry to become machine processable activities, e.g., identifying changes to the content of individual package inserts or comparing information across a broad range of SPL instances.

[38] For example, XML is supported by all major Internet browsers.

[39] SPL embodies syntactic interoperability through the use of XML schemas and semantic interoperability (i.e., unambiguous interpretation of the XML tags in SPL) through derivation from the HL7 RIM model. See http://www.hl7.org for more information or http://www.healthcare-informatics.com/webinars/05_20_04.htm for a web-based discussion of HL7 and semantic interoperability.

[40] This converse may also be supported, i.e., SPL could be populated with information from other information systems (rather than supplying information), ensuring data integrity across data systems. However, in either role (i.e., as receiver or supplier of 'data'), SPL serves as an envelope for this information.

7.1.3 Human Readability & Data Presentation

SPL expresses the content of drug labeling separately from its presentation. A uniform presentation (or rendering) of content across all package inserts can be supported by FDA; this is an important benefit to potential end-users of SPL content.

Equally valuable is the ability to have customized presentations (renderings) for end users and within regulatory agencies and industry. Such renderings may exist for a single SPL instance or across multiple SPL documents. Customized rendering should permit use of the content of pharmaceutical labeling by both the health care industry and regulatory agencies in medically and regulatory important ways that were previously inaccessible.

As an XML-based document, an SPL document retains human-readability in its native document form; sections of SPL can be easily comprehended, displayed, and edited without transformation or special rendering.

7.1.4 Streamlined Processing

SPL has significant advantages for internal processing of documents. XML-based documents permit separation of content for different purposes, e.g., for generating different regulatory submissions by pharmaceutical companies or for entry into different database systems by FDA, e.g., drug listing. SPL itself can be directly constructed via database systems. Modular use of document sections promotes consistency of content in company-generated documents and the reduction or elimination of redundancies; this similarly aids FDA in maintaining consistency across documents.

SPL facilitates the comparisons of document content (both narrative and data elements) across documents or between current and earlier versions of the same document. Document revision histories can be automated and viewed.

The use of a structured standard would permit processes developed by one organization to be used seamlessly by others; for examples, objects to 'test' whether an SPL meets certain regulatory criteria (e.g., CBE) developed by a regulatory agency could be distributed for use by pharmaceutical companies. (Similar functionality could also be provided as web services.)

SPL also permits automated insertion and deletion of sections while confirming and maintaining the data integrity of non-altered content.

7.1.5 Scalability & Flexibility

The SPL format permits scalable implementation of document markups that may be of varying complexities. Such markups are flexible, i.e., naming or nesting of sections are not imposed upon the document, and permits integration of XML and non-XML sources with common metadata. Lastly, multiple data types (e.g. images, tables, and/or text) may be embedded into SPL.

As a schema-based standard, evolution of SPL can occur in a backward-compatible fashion that maintains compatibility with earlier versions of the standard. Similarly, transformations of XML documents can permit third parties to easily convert non-SPL defined information into clinical products and data sources while insuring integrity of the SPL content.

7.1.6 Tool independence

SPL as a non-proprietary open standard is compatible with a wide range of XML document creation applications while remaining independent of underlying storage mechanisms and/or document management systems. It is available to vendors who can apply the standard to any XML-based product they currently market. Vendors can build new tools or modify any current tools they may have to work with the standard.

Each business may define their own processes to incorporate SPL and choose the tools which will best complement the way they do their work, recognizing that SPL documents will be compatible with any other system.

The development of SPL as an open standard within the HL7 standards development process ensures that future development of the standard reflects a consensus process among industry, regulatory authorities, and the general health care community.

Use of SPL as an unambiguous, standardized markup (including use of standard controlled vocabularies and coding systems) should ultimately facilitate use of the content of a text document as data, permitting accurate database-like searches across SPL documents to improve patient care and safety. With data in SPL format, there may be opportunities to eliminate redundant data collection and manual processing that are used in other submissions, such as drug listing. Additionally, process efficiencies may surface with other organizations which use package insert information.

The potential benefits of SPL apply to every stakeholder who implements SPL. Even without the obvious benefit for use within FDA, benefit to other organizations that will serve as information-providers (such as the National Library of Medicine [NLM]), and information-consumers such as health care professionals and the general public make the transition to SPL by FDA necessary.

7.2 SPL Standard Stylesheet and FDA Implementation of Stylesheets[41]

7.2.1 Introduction

SPL is a form of document markup that is based on the HL7 clinical document architecture that will enable the standardization of the structure and semantics of the required content of drug product labeling as described in FDA regulations and guidance. While the SPL document is human readable, there are no formatting structures defined under the SPL-schema definition per se: formatting requirements specific for individual SPL instances are communicated via the *styleCode* attribute. The process of generating a formatted human readable display (or presentation of an SPL document [instance]) is referred to as rendering[42] ; for SPL, this means having default display characteristics modified when needed by the *styleCode* attribute value.

This SPL standard stylesheet is used for the display of an SPL document. It consists collectively of two files, an SPL extensible stylesheet language file (currently spl-2a.0.xsl) and an SPL cascading stylesheet file (currently spl-2a.0.css). These files are W3C standard-compliant file types that define the baseline set of style conventions for viewing an SPL XML file in a current browser, such as Microsoft Internet Explorer 6.0, Mozilla 1.7, Netscape Navigator 7.1, or Opera 7.54 or greater.[43] File spl-2a.0.xsl transforms an SPL document to a W3C standard-compliant XHTML document. spl-2a.0.css defines cascading stylesheet classes for rendering the XHTML-transformed SPL document. Both files were created to be "browser-independent," meaning software products that support the W3C standards should render a valid SPL document similarly.[44]

Examples of rendering via the SPL standard stylesheet are shown in Sec. 4.1.4

If additional styles are needed for the proper display of the FDA content of labeling that have not been adopted in the standard stylesheet, these will be included in a FDA-specific stylesheet that will add, or cascade, with the standard stylesheet. The file is named similarly to spl-2a.0.css, i.e., spl-fda-2.0.css.

Spl-2a.0.xsl, spl-2a.0.css, and spl-fda-2.0 (if this file exists) will be available at http://www.fda.gov/oc/datacouncil/spl.html. At the present time there is no file spl-fda-2.0.css as there is no need for this file at the present time.

It should be noted that the css files are coded in the xsl file, i.e., in the actual coding of the SPL file the names of the active css files are unnecessary. Specifying the current xsl file in the SPL document will automatically select the css files current at the time the SPL is created.

7.2.2 SPL Stylesheet Components

The files collectively referred to as the SPL Standard Stylesheet (i.e., the separate xsl and css files) define the baseline set of style conventions for viewing an SPL-compliant XML file in a browser.

To illustrate how they are used, the following code represents a snippet from the gemcitabine demonstration SPL document cited earlier:

---

[41] As noted earlier, the current 'SPL 2a' version that this implementation guide addresses uses an FDA-specific schema that has not been approved by HL7. SPL Release 1 contained an 'HL7 standard stylesheet (spl-1.0.xsl)'; the stylesheet to support this release, spl-2a.0.xsl , located at http:// www.fda.gov/oc/datacouncil/spl.html, is not equivalent to the HL7 stylesheet. Any reference to an 'HL7 standard stylkesheet' in the discussion below should refer to the FDA-supplied stylesheet. It is the intention that if SPL Release 2 passes membership ballot, an HL7 standard stylesheet will be made available.

[42] In this discussion, 'presentation,' 'display,' and 'rendition' are used synonymously.

[43] Support for a W3C-complaint XML stylesheet transformation is not necessarily for properly rendering SPL. Server-based implementations of SPL may transform SPL via the standard XSL file and transmit the transformed XHTML document for rendering on a local device. In this model, only software support for rendering an XML/XHTML document with stylesheet classes is necessary for display.

[44] The end-user display/presentation is the XHTML transformed SPL document. Direct viewing of the SPL document, although possible via a specific CSS file, is not supported or recommended.

```
<component>
  <section>
    <id  root=" A856E13A-8AA7-4C45-B378-97957CDFFC81"/>
    <code code="34090-1" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
          displayName="Clinical pharmacology"/>
    <title>CLINICAL PHARMACOLOGY</title>
    <component>
      <section>
      <id root="DD761815-06CE-47CA-A2D2-EFB2F24EFA44"/>
        <text>
        <paragraph>
          Gemcitabine demonstrated dose-dependent synergistic activity with cisplatin <content
          styleCode="italics">in vitro</content>. No effect of cisplatin on gemcitabine triphosphate
          accumulation or DNA double-strand breaks was observed. <content
          styleCode="italics">In vivo</content>, gemcitabine showed activity in combination with
          cisplatin against the LX-1 and CALU-6 human lung xenografts, but minimal activity was
          seen with the NCI-H460 or NCI-H520 xenografts. Gemcitabine was synergistic with
          cisplatin in the Lewis lung murine xenograft. Sequential exposure to gemcitabine 4 hours
          before cisplatin produced the greatest interaction.
        </paragraph>
        ……
        ……
```

These excerpts of SPL-compliant XML are transformed by the following code in spl-2a.xsl[45]:

```
…..
    <xsl:template match="v3:section/v3:title">
      <xsl:param name="sectionLevel" select="count(ancestor::v3:section)"/>
      <xsl:element name="h{$sectionLevel}">
        <xsl:apply-templates select="@*"/>
        <xsl:apply-templates mode="mixed" select="node()"/>
        </xsl:element>
    </xsl:template>
  …..

    <xsl:template match="v3:paragraph">
    <p>
      <xsl:apply-templates select="@*|v3:caption"/>
      <xsl:apply-templates mode="mixed"
          select="node()[not(self::v3:caption)]"/>
    </p>
    </xsl:template>
  …..

  <xsl:template mode="mixed" match="v3:content">
    <span>
      <xsl:apply-templates mode="mixed" select="@*|node()"/>
    </span>
  </xsl:template>
  <xsl:template mode="mixed" match="v3:content[@styleCode='yes']">
    <em><xsl:apply-templates mode="mixed" select="@*|node()"/></em>
  </xsl:template>
  <xsl:template mode="mixed" match="v3:content[@styeCode='italics']">
    <em class="Italics{styleCode}">
      <xsl:apply-templates mode="mixed"
        select="@*[not(local-name()='styleCode')]|node()"/>
    </em>
```

[45] This code is meant only to be illustrative and does not represent the current stylesheet content.

```
</xsl:template>
.....
```

into this HTML-compliant markup:

```
<h1>CLINICAL PHARMACOLOGY</h1>
<p>
Gemcitabine demonstrated dose-dependent synergistic activity with cisplatin <em class="Italics">in vitro</em>. No effect of cisplatin on gemcitabine triphosphate accumulation or DNA double-strand breaks was observed. <em class="Italics">In vivo</em>, gemcitabine showed activity in combination with cisplatin against the LX-1 and CALU-6 human lung xenografts, but minimal activity was seen with the NCI-H460 or NCI-H520 xenografts. Gemcitabine was synergistic with cisplatin in the Lewis lung murine xenograft. Sequential exposure to gemcitabine 4 hours before cisplatin produced the greatest interaction.
</p>
```

The HTML <p> element is rendered according to the CSS classes defined in the css file. For example, the <p> element is displayed by this class definition:[46]

```
p {
    text-indent: 0em;
    margin-top: 0em;
    margin-bottom: 0ex;
    line-height: 2.2ex;
}
```

with default styles inherited from the body element:

```
body {
    background-color: white;
    color: black;
    font-family: Arial Unicode MS
    margin-left: 4em;
    margin-right: 4em;
    }
```

Therefore, in a compliant browser supporting CSS formatting, the snippet above would be presented as black text on a white background. There would be 4 cm left and right margins, and a line-height of 2.2 ex.

When displayed in a W3C complaint browser, the text would appear similar to the following[47]:

CLINICAL PHARMACOLOGY

Gemcitabine demonstrated dose-dependent synergistic activity with cisplatin *in vitro*. No effect of cisplatin on gemcitabine triphosphate accumulation or DNA double-strand breaks was observed. *In vivo*, gemcitabine showed activity in combination with cisplatin against the LX-1 and CALU-6 human lung xenografts, but minimal activity was seen with the NCI-H460 or NCI-H520 xenografts. Gemcitabine was synergistic with cisplatin in the Lewis lung murine xenograft. Sequential exposure to gemcitabine 4 hours before cisplatin produced the greatest interaction.

[46] Note: the unit 'em' is a measure of horizontal space (width) that depends on the currently effective font (approximately the width of the lower case m. The unit 'ex' is a measure of vertical space (height) that depends on the currently effective font (approximately the height of a lower case x). For example, the 4 em value indicates that for the margin-left and margin-right properties, the left and right margins will be four times the height of the default font rendering the document. The sans-serif value for the font-family property specifies the default non-serif font for the browser, e.g., arial or helvetica fonts.

[47] The margins are inaccurate in this example, having been reformatted for this specific document.

The cascading stylesheet defines a default CSS class for each XHTML element resulting from the transformation of SPL to XHTML. Additional formatting choices are available to render SPL content in specific ways via the *styleCode* attribute. If necessary, these additional formatting choices (classes) would be available in the FDA SPL stylesheet for the display of the FDA content of labeling (see the next section for more details), For example, the following SPL snippet:

```
……
    <tr styleCode="Botrule">
        <td>&amp;#160;&amp;#160;Range</td>
        <td>36 to 88</td>
        <td>35 to 79</td>
        <td>33 to 76</td>
        <td>35 to 75</td>
    </tr>
    <tr>
        <td>Stage IIIA</td>
        <td>7%</td>
        <td>7%</td>
        <td>N/A</td>
        <td>N/A</td>
    </tr>
    ….
```

will render as two rows in the following table:[48]

Table 2: Randomized Trials of Combination Therapy with Gemzar plus Cisplatin in NSCLC

Trial	28-day Schedule[a]			21-day Schedule[b]		
Treatment Arm	Gemzar/ Cisplatin	Cisplatin		Gemzar/ Cisplatin	Cisplatin/ Etoposide	
Number of patients	260	262		69	66	
Male	182	186		64	61	
Female	78	76		5	5	
Median age, years	62	63		58	60	
Range	36 to 88	35 to 79		33 to 76	35 to 75	
Stage IIIA	7%	7%		N/A	N/A	
Stage IIIB	26%	23%		48%	52%	
Stage IV	67%	70%		52%	49%	
Baseline KPS[c] 70 to 80	41%	44%		45%	52%	
Baseline KPS[c] 90 to 100	57%	55%		55%	49%	
Survival			p=0.008			p=0.18
Median, months	9.0	7.6		8.7	7.0	
(95%,C.I.) months	8.2, 11.0	6.6, 8.8		7.8, 10.1	6.0, 9.7	
Time to Disease Progression			p=0.009			p=0.015
Median, months	5.2	3.7		5.0	4.1	
(95%,C.I.) months	4.2, 5.7	3.0, 4.3		4.2, 6.4	2.4, 4.5	
Tumor Response	26%	10%	p<0.0001[d]	33%	14%	p<0.01[d]

[a] 28-day schedule – Gemzar plus cisplatin: Gemzar 1000 mg/m 2 on Days 1, 8, and 15 and cisplatin 100 mg/m 2 on Day 1 every 28 days; Single-agent cisplatin: cisplatin 100 mg/m 2 on Day 1 every 28 days.

[b] 21-day schedule – Gemzar plus cisplatin: Gemzar 1250 mg/m 2 on Days 1 and 8 and cisplatin 100 mg/m 2 on Day 1 every 21 days; Etoposide plus Cisplatin: cisplatin 100 mg/m 2 on Day 1 and I.V. etoposide 100 mg/m 2 on Days 1, 2, and 3 every 21 days.

[c] Karnofsky Performance Status.

[d] p-value for tumor response was calculated using the two-sided Fisher's exact test for difference in binomial proportions. All other p-values were calculated using the Logrank test for difference in overall time to an event.

In this example, specifying "Botrule" for the styleCode attribute to the first <tr> element creates a horizontal rule below the first row in the table (bottom rule)

The complete list of available CSS formatting classes is discussed in Section 4.1.5, Tables.

[48] In a browser supporting CSS1 and certain class properties available in CSS2.

7.2.3 Creation and Use of SPL Stylesheets

To the extent possible, users should adopt the CSS classes available in the standard css stylesheet for creating SPL documents. If specific requirements necessitate additional stylesheet classes to accommodate the proper display of a valid SPL document, HL7 permits authorized agencies to define a local cascading stylesheet that extends the existing standard stylesheet.

FDA is responsible for defining the local SPL practices for regulatory submissions in the United States. At the present time, no additional spl classes have been defined. When (and if) these become necessary, the additional classes available for SPL submissions in the U.S. (spl-fda-2.0.css) will be located at http://www.fda.gov/oc/datacouncil/spl.html By cascading spl-fda-20.css with spl-2a.css, users have access to a superset of CSS formatting classes beyond that available in spl-2a.css alone. No classes are defined in spl-fda-2.0.css that override existing classes in spl-2a.css, meaning that no identical class names are used in spl-fda-2.0 and spl-2a.css. The addition of local classes to spl-2a.css provides a means for supplementing the standard stylesheet in those circumstances for which this is necessary. Local classes are not intended to fundamentally alter the appearance of a rendered SPL document as defined by the standard stylesheet through, for example, extensively modifying common elements, such as the <paragraph> or <list> elements.

FDA will maintain spl-fda-2.0. Use of local classes permits FDA to act independently from HL7 when the rapid addition of CSS classes is necessary to meet immediate needs. FDA will forward any added classes to the HL7 for consideration in adding to the SPL standard stylesheet with the goal of minimizing locally (i.e., FDA) defined classes. All added classes adopted in the SPL standard stylesheet will be removed from the locally defined stylesheets.

Formatting classes not present in the SPL standard stylesheet (spl-2a.css) or in FDA-specific additional classes defined in spl-fda-2.0.css will not be supported by FDA and should not be used in SPL documents submitted to FDA.

The SPL transformation stylesheet associated with HL7-balloted releases is maintained by HL7 and is designed to render an XHTML-compliant document formatted for human readability according to standard document constructs. It is anticipated that virtually all formatting accommodations necessary for rendering SPL will be accommodated though HL7 or realm-specific CSS classes as opposed to alterations in the transformation (xsl) stylesheet.

7.3 Glossary

Term or Abbreviation	Definition
21CFR201.56	Code of Federal Regulations, Title 21, Federal Food, Drug and Cosmetic Act, Part 201.56, "General Requirements on content and format of labeling for human prescription drugs." Access via: http://www.gpoaccess.gov/cfr/index.html
21CFR201.57	Code of Federal Regulations, Title 21, Federal Food, Drug and Cosmetic Act, Part 201.57, "Specific Requirements on content and format of labeling for human prescription drugs." Access via: http://www.gpoaccess.gov/cfr/index.html
Active ingredient	A substance responsible for effects of a medicinal product defined by the complete molecular structure (e.g., with any counter ion) or, if the precise molecular structure is not known, by an unambiguous definition (e.g., the process used to create the substance), e.g. codeine phosphate. Active Ingredients in SPL are modeled by the <activeIngredient> data element described in Table 15. In HL7, active ingredient is strictly defined as a role representing a therapeutically active ingredient (player) in a mixture (scoper), where the mixture is typically a manufactured pharmaceutical.
Active moiety	An active moiety in a product is defined in 21CFR314.108: "Active moiety means the molecule or ion, excluding those appended portions of the molecule that cause the drug to be an ester, salt (including a salt with hydrogen or coordination bonds), or other noncovalent derivative (such as a complex, chelate, or clathrate) of the molecule, responsible for the physiological or pharmacological action of the drug substance." Active moiety is included in SPL as a child element (activeMoiety) of the Active ingredient element and is described in Table 15. As discussed below, this is also coded in SPL by a UNII (Unique Ingredient Identifier provided by FDA.
ANSI	American National Standards Institute
Approved Labeling	Labeling content that has been officially approved by the FDA. This does not refer to labeling content that is implemented by the sponsor (and not formally FDA approved) through CBE or annual report changes, etc.
ASCII	American Standard Code for Information Interchange, a common 8-bit character encoding of printed (and some non-printed characters) for electronic communication.
Attribute	A name-value pair included inside an XML element tag, e.g., <id root="2 F33776B3-2DC8-435B-856B-444DD69F6CD7"/> where 'id' is the element name and 'root' the attribute.
CDA	HL7 Clinical Document Architecture
Clinical document	An HL7 clinical document is a documentation of clinical observations or other medically relevant information with the following characteristics: • Persistence – A clinical document continues to exist in an unaltered state, for a time period defined by local and regulatory requirements. • Stewardship – A clinical document is maintained by a person or organization entrusted with its care. • Potential for authentication – A clinical document is an assemblage of information that is intended to be legally authenticated. • Wholeness – Authentication of a clinical document applies to the whole and does not apply to portions of the document without the full context of the document. • Human readability – A clinical document is human readable. In HL7, Clinical Documents are derived from the Clinical Document Architecture (see below). SPL is derived from the HL7 Clinical Document Architecture.
Clinical Document Architecture	ANSI/HL7 CDA R1.0-2000. Specification for the structure and semantics of "clinical documents" for the purpose of exchange. Defined by HL7. See http://www.hl7.org/lib_admin/docs.cfm?dir=library\committees\structure&comm=structure for a copy of the CDA standard; additional information is available at http://www.hl7.org The architecture for structured documents defines relationships between documents and

Term or Abbreviation	Definition
	document specifications in terms of specialization and inheritance.
Content of labeling	All text, tables and figures in labeling as described in regulations for a specific product (e.g., 21 CFR 201.56 and 201.57 for human prescription drugs, 201.66 for human over-the-counter drugs). For SPL, content of labeling includes sections such as the Patient Package Insert (PPI) or MedGuide if these are included as part of the printed package insert.
DEA	
Dosage Form	1) Form of product, (e.g., tablet, capsule, solution, etc.) that contains an active drug ingredient or placebo. Dosage Form is modeled as an Observation in SPL using the FDA CDER Data Standards Table (see Table 15.) 2) A finished dosage form as described in regulations
Drug	1) Articles recognized in the Official United States Pharmacopoeia, Official Homeopathic Pharmacopoeia of the United States, or Official National Formulary 2) Articles intended for the use in the diagnosis, cure, mitigation, treatment, or prevention of disease in man or other animals
Electronic Labeling Rule	"Requirements for Submission of Labeling for Human Prescription Drugs and Biologics in Electronic Format",Published in the Federal Register: December 11, 2003, Volume 68, Number 238. (http://frwebgate1.access.gpo.gov/cgi-bin/waisgate.cgi?WAISdocID=154453135555+1+0+0&WAISaction=retrieve) Guidance for Industry regarding the Electronic Labeling Rule is available at: http://www.fda.gov/ohrms/dockets/98fr/2004d-0041-gdl0001.doc
Element	A section of text in an XML document delimited by start and end tags; or in the case of empty elements (elements with no content, only attributes), indicated by an empty tag. See 7.5, XML Primer.
Entity	A representation of a special character that cannot be used in XML coding of text, e.g., "<" must be coded as "<" if used within a paragraph to avoid confusion with element tags. Entities are described in Table 5.
ELIPs	FDA's Electronic Labeling Information Processing System.
Established name	1) The official name of a drug substance 2) In US FD&C Act, the term "established name", with respect to a drug or ingredient thereof, means (A) the applicable official name designated pursuant to section 508 of the FD&C Act, or (B), if there is no such name and such drug, or such ingredient, is an article recognized in an official compendium, then the official title thereof in such compendium, or (C) if neither clause (A) nor clause (B) of this subparagraph applies, the common or usual name, if any, of such drug or of such ingredient, except that where clause (B) of this subparagraph applies to an article recognized in the United States Pharmacopoeia and in the Homoeopathic Pharmacopoeia under different official titles, the official title used in the United States Pharmacopoeia shall apply unless it is labeled and offered for sales as a homoeopathic drug, in which case the official title used in the Homoeopathic Pharmacopoeia shall apply.
FD&C Act	Food, Drug and Cosmetic Act – http://www.fda.gov/opacom/laws/fdcact/fdctoc.htm
FDA	Food and Drug Administration.
Granularity	The relative size of a defined 'semantic' or informational unit; in the context of this specification, granularity refers to the size of an information unit where <section> would be coarse grained and a data point would be fine grained. The degree of granularity in SPL is defined by the ability of the author to define related information into <sections> of the SPL. Increased granularity is recommended by dividing <section>s into smaller sub<section>s where each subsection contains related information. See section 4.1.2 for a discussion of this in the context of SPL.
GUID	A Globally Unique Identifier (GUID) used to identify an element, that is, a unique value that does not exist in any other SPL (or in any other context where GUID are used). See Sections 4.1.3.1 and 7.9.6 for additional discussion.

Term or Abbreviation	Definition
Harmonization	The HL7 process that ensures consistency between all HL7 messages and documents, including SPL.
Health Level Seven	An ANSI-accredited States Developing Organization (SDO) operating in the healthcare arena. "Level Seven" refers to the highest level of International Standards Organization's (ISO) communications model for Open Systems Interconnection (OSI) – the application level. The application level addresses definition of the data to be exchanged, the timing of the interchange and the communication of certain errors to the application. The seventh level supports such functions as security checks, participant identification, availability checks, exchange mechanism negotiations and, most importantly, data exchange structuring. (See http://www.hl7.org)
HL7	See Health Level Seven above. The "7" stands for the Application level of the ISO communication model – ISO level 7.
HTML	Hypertext Markup Language, a specification of the W3C (World Wide Web Consortium) that provides for the markup of documents for display in a web browser, i.e., the specification for how HTML documents should be written for display in a web browser. HTML has been supplanted by a W3C specification for XHTML, an XML-based specification for HTML documents. (See http://www.w3c.org)
IETF	The IETF is the protocol engineering and development arm of the Internet.
Inactive ingredient	Any component of a drug product other than an active ingredient, as described in regulations
Ingredient	Any component of a drug product
Label	Label is considered identical to the FDA-defined package insert or Final Printed Labeling (FPL).
Legal authentication	A completion status in which a document has been signed manually or electronically by the individual who is legally responsible for that document.
LOINC	LOINC® (Logical Observations, Identifiers, Names, and Codes, http://www.regenstrief.org/loinc.htm). LOINC® is a coding system maintained by the Regenstrief Institute for Health Care primarily for coding and electronic transmission of laboratory and other medical information. LOINC codes within SPL identify specific values, e.g., sections within the FDA package insert (see Appendix 7.6.)
Markup	Computer-processable annotations within an XML document not meant to be displayed but needed to structure the document, e.g., element names such as <section>.
NDC	Each drug product listed under Section 510 of the Federal Food, Drug, and Cosmetic Act is assigned a unique 10-digit, 3-segment number. This number, known as the National Drug Code (NDC), identifies the labeler/vendor, product, and trade package size. The first segment, the labeler code, is assigned by the FDA. A labeler is any firm that manufactures, repacks or distributes a drug product. The second segment, the product code, identifies a specific strength, dosage form, and formulation for a particular firm. The third segment, the package code identifies package sizes. Both the product and package codes are assigned by the firm. The NDC will be in one of the following configurations: 4-4-2, 5-3-2, or 5-4-1. See http://www.fda.gov/cder/ndc/ for additional information.
National Library of Medicine	(NLM) is the world's largest biomedical library. The Library acts as a supplemental resource after local, regional, and other national resources have been contacted (such as the FDA). NLM also creates and maintains databases (including all US marketed products or Approved labeling) and databanks for general use.
Package	Container for holding a product, as described in US FDA regulations. Package in SPL is modeled in the <containerPackage> element. (See Table 15.)
Package Insert	See 'Label' above.
Package quantity	The net quantity of contents of a product in package form
Packaged Medicinal Product	A Packaged Medicinal Product is a Proprietary Medicinal Product (see below) in a specific package/quantity, i.e., a Packaged Medicinal Product will have an associated NDC code.
Patient Package Insert	The package package insert is an additional document included with prescription product distribution at the point of sale. Information from a patient package insert may be

Term or Abbreviation	Definition
	appended to labeling in which case it is content of labeling and must be included with SPL. Similar considerations apply to MedGuides.
Proprietary Medical Product	Proprietary Medicinal Product: includes one or more established medicinal products along with inactive ingredients and appearance and manufacturer or marketing authorization holder name, country and registration number, proprietary name, established number, marketing status, drug product type, and distributor. This is reflected by the proprietary medicinal code value under Proprietary name. An example would be Amoxil 500 mg tablet. There can be multiple packages (e.g., NDC codes) for a given proprietary medicinal package. A *Packaged Medicinal Product* is a Proprietary Medicinal Product in a specific package/quantity, i.e., a Packaged Medicinal Product will have an associated NDC code.
Proprietary name	1) A name that a company uses for the commercial distribution of a drug product; may also be known as the brand name 2) The brand name as described in regulations
Realm	The geographical, organizational, or political environment where the HL7 standard is being used. For SPL, the US realm is used as this reflects a United States regulatory document.
Reference Information Model (RIM)	An information model used as the base model for deriving all HL7 messages and clinical documents. Since all HL7 documents are derived from a common model, interoperability and consistency between documents is assured.
Regulated product	A drug product that is subject to regulatory requirements.
RIM	Reference Information Model. The RIM is the basis for which all HL7 documents and messages are derived. By forcing all HL7 documents and messages to derive froma single source, all HL7 messages are compatible.
Schema	A formal definition of the structure and content of a type of XML document. An XML schema is identified by the *.xsd extension of an XML file. Every SPL document must be be valid against the SPL schema, e.g., conforming to the definition of the structure for an SPL document.
Section	The <section> element identifies units of information within SPL and can include nested subsections. See Section 4.1.
SGML	Standard Generalized Markup Language, ISO 8879:1986(E) as amended and corrected. XML is derived from SGML.
SPL body	The SPL Body is a section containing the data elements and content of labeling.
SPL header	The SPL Header identifies and classifies the document and may provide information on the owner of the marketing authority for the product, the author, legal authenticator, and reviewers
SPL Instance	An SPL instance is a specific SPL document, e.g., a SPL for singulair.
SPL specification	The Structured Product Labeling (SPL) specification is a document markup standard that specifies the structure and semantics for the regulatory requirements and content of product labeling.
SRS	The FDA Substance Registration System, the source for UNII codes used in SPL (see UNII below)
Strength	1) The concentration of a substance (e.g., the concentration of drug in a dosage form) 2) The measurement of the drug substance as described in regulations.
Stylesheet	A file that describes how to display an XML document of a given type. See XSL below.
UNII	Unique identifier codes for active and inactive ingredients. These coded values are provided by FDA. UNII's are used as codes in the <code> element child of the <activeIngredient>, <inactiveIngredient> and <activeMoiety> elements.
UUID	Universally Globally Unique Identifier, an alternative name for GUID's (see above).
Valid document	A document which meets all of the validity constraints in the XML Specification. In SPL this is a document that is well formed (follows the rules of XML) and is valid, i.e., follows the rules of the SPL schema. See 7.5.8, “Well formed and Valid XML Documents” in the XML Primer.

Term or Abbreviation	Definition
Value set	In HL7, a vocabulary domain that has been constrained to a particular realm and coding system.
Vocabulary domain	In HL7, the set of all concepts that can be taken as valid values in an instance of a coded field or attribute
W3C	The World Wide Web Consortium, an international industry consortium (http://www.w3.org)
W3C Schema	The three-part schema specification issued by the W3C XML Schema Part 0: Primer, W3C Recommendation 2-May-2001, http://www.w3.org/TR/xmlschema-0/ XML Schema Part 1: Structures, W3C Recommendation, 2-May-2001, http://www.w3.org/TR/xmlschema-1/ XML Schema Part 2: Datatypes, W3C Recommendation, 2-May-2001, http://www.w3.org/TR/xmlschema-2/
XML	Extensible Markup Language, specification of the W3C, a formal subset of SGML (http://www.w3.org/TR/REC-xml). See Appendix 7.5, XML Primer.
XML declaration	Markup stating that that document is an XML document and stating to which version of the XML specification the document is conformant.
XML document	SPL is an XML document. An XML consists of a prolog (processing instructions), a root document element, and other objects. See Appendix 7.5, XML Primer.
XML schema	See W3C Schema
XSL	Extensible Style Language, a specification of the W3C (www.w3.org/Style/XSL/). An XSL stylesheet in SPL collectively consists of several files, a transformation file (e.g., spl.xsl) and a cascading style sheet (e.g., spl.css). These files 'transform' SPL into an HTML files that can be viewed by a standard Web browser. See Appendix 7.2, SPL standard stylesheet.
XSLT	XSL transformation language, a specification of the W3C (http://www.w3.org/TR/xslt). XSLT is one part of what is collectively considered the XSL stylesheet (see XSL above).

7.4 SPL Image File Types

The following guidelines should be used for the inclusion of image files in SPL submissions.

7.4.1 File types:

JPEG (*.jpg) and GIF (*.gif) file formats can be used during the draft implementation of SPL. JPEG files are preferred; GIF file types should only be used if a JPEG alternative does not exist. (It is expected this will be extremely uncommon.)

7.4.2 File names:

It is required that all file names cited inside SPL should match exactly the actual file names. For example:

```
<ObservationMedia ID="MM1">
    <value xsi:type="ED" mediaType="image/jpeg">
        <reference value="gemzar_structure.jpg"/>
    </value>
</ObservationMedia>
```

All files in an SPL submission should be at the same base level. No relative or absolute paths should be used in graphic file references in SPL. For example, <reference value="./image_files/gemzar_structure.jpg"/> should not be used but <reference value="gemzar_structure.jpg"/> is acceptable. Graphic fle references should also not contain URLs containing scheme identifiers (file: or http:) ; all graphic images must be submitted with the SPL document.

7.4.3 Style Rules:

The following style rules must be applied to the use of image file names in SPL:

1. All file names should be in lower case only (no mixed case).
2. File names should be simple and use only the brand name as a prefix.
3. No spaces in file names.
4. Image file names have the same prefix as the main SPL file, followed by a dash and a short suffix to distinguish it, e.g., toprol-xl-01.jpg, toprol-xl-02.jpg, toprol-xl-structure.jpg, or toprol-xl-figure-01.jpg
5. File extensions should be restricted to 3 characters, i.e., JPEG files should be .jpg not .jpeg or .jpe.
6. Image or other files should not be placed into nested directories. The SPL file and all its other files should be at the same single directory level.

7.5 XML Primer

7.5.1 Introduction

XML (Extensible Markup Language) is a method for structuring electronically transmitted information; however, it does not specify the structure of the document itself. Instead, XML standards provide a mechanism that permits individual rules to be written for specific document types; one such document type is SPL or Structured Product Labeling. SPL is a specific type of XML document that is defined by a schema, or rules, to create this type of document. The schema for a specific type of document (or the document itself) is not part of the XML standard but must be written following rules of the XML standard. This concept is analogous to FDA regulations or a CTD, in that the regulations and CTD define the content and format a New Drug Application or a Biologics License Application, but do not specify the actual content of any specific application.

The schema or 'rules' list and define the components that make the document XML-compliant. The rules for a specific type of XML document define the structure for the information in that particular document type, e.g., an ‘address' for one specific document type may be specified to consist of a 'street name', a 'city', and a 'zip code'. The zip code may be further specified as only a 5 digit number.[49] In more technical terms, the rules of an XML document type, or the 'schema' for that particular type, unambiguously specify the 'syntax' that a document can take.

Information in an XML document may be intended for a display to be read by humans through web browsers, while other XML documents contain primarily 'data' that is meant for receipt and processing by another computer. In both cases XML provides the structure so both the 'sender' and the 'receiver' can understand the information, regardless of whether the 'receiver' is a human or another computer. XML documents, such as SPL contain both types of information in one document, i.e., the 'content of labeling' is meant for display and reading by humans, and 'data elements' which are meant to be 'interpreted' by computers and used in database systems where they may be subsequently converted for presentation to a human reader.

The following is an abbreviated introduction to basic XML terminology used in this implementation guide. Should you wish to learn more about XML, Internet or printed resources are readily available.

7.5.2 Elements and Tags

An element is the basic building block of an XML document. Element names are defined by the author of the the schema for the document; in some cases they may be standardized (e.g., elements used for web pages, SPL elements based on the HL7 information model). The basic format of an XML element is: <tag>information</tag> (i.e., a set of tags wraps a particular piece of the information being structured), as shown in the following example:

```
<model>Ferrari 360 Modena</model>
```

Every XML element has a start tag (<model> in this example), an end tag </model>, and the element content ("Ferrari 360 Modena "). A start tag always has a left angle bracket (<), the element name (model), and a right angle bracket (>). The end tag is identical to the start tag with the addition of a slash (/) after the left angle bracket to indicate “close tag”.[50] Start tags and end tags are considered *markup*, i.e., the information in an XML document that is not meant to be displayed but identifies the structure of the document. Processing instructions, as well as comments (both described later in this section) are other examples of markup.

[49] The rules could have just as easily specified a 9 digit zip code -- the form an XML document takes is dependent on the rules that were written for that document type by the author of the document type.

[50] Because of this, a left angle bracket (<) cannot be used in the content of an XML element since this will be misinterpreted as the start of a new element. To include a left angle bracket as an in SPL content an entity reference must be used, e.g., <model>Ford &#lt;car &#gt;</model> which would be displayed as "Ford <car>" in a browser. See Section 4.1.4.3 for other examples.

A schema consists of defined elements usually specified to be in a particular order, possibly with other elements nested inside them. (This is not always true: a schema can define a collection of elements below the root element that can exist in potentially different, or even random, orders.) The root element is the first element in an XML document. All the other elements in an XML document are nested inside the root element, also referred to as being children of the root element. Each child element may consist of other elements, content information (i.e., text), or a combination of text and other elements. A special type of element is the empty element which may contain attributes (see below) but neither text nor other elements. An empty tag has a special form where there is no close tag (essentially a start and end tag combined). <emptyTag/> is an example of the form an empty tag would take.

Since the first element in an XML document is always the root element, e.g., <document>, the last tag is always the end tag for the root element, e.g., </document>. Element names (and therefore tags) are case sensitive, i.e., <document> is a different tag from <Document>.[51]

For example, if *car* is the "parent" or root element of an XML document, sub-elements or children of *car* might include *make, model* and *color* so, in XML markup, the car: a red 360 Modena Spider would appear as:

```
<car>
        <make>Ferrari</make>
        <model>360 Modena Spider</model>
        <color>RED</color>
        <vin vinNumber="12335566699"/>
</car>
```

car is an element which contains only other elements; these elements are the make, *model, color,* and *vin* elements. Another way of expressing this is that make, model, color and vin are children of the car element, or 'nested' within the car element. 'vin' is an empty element with the attribute vinNumber (see below).[52]

7.5.3 Attributes

Attributes are name/value pairs that are associated with a particular element; they may further define or modify the element. Attributes appear inside start tags or empty tags. Attributes must have a "value" (including a possible default value) assigned; this value will always be enclosed by quotes.[53] The name/value pair is always expressed as:

```
<element attributeName="value">content</element>
```

For example, one method to identify a person's age via an attribute in an XML document could be:

```
<person age="18">Toni</person>
```

[51] The distinction between elements and tags is frequently confusing and some people use them interchangeably. In the example just presented, *car* is an element, with child elements *make*, *model*, etc. Another element in the example below is *make*, where the value is Ford and is identified by <make>Ford</make>. A tag is specific markup in an XML document, e.g., </make> is an end tag for the element make. Elements may be referred to simply by their name whereas a tag always has brackets, e.g., <car> is the start tag for the car element. However, a common convention in narrative about XML is to surround element names with angle brackets (this convention was used in the SPL specification). If the example of *car* above were a complete XML document (rather than being part of another document), then *car* would be the root element of the document.

[52] An important point is illustrated by comparison of the element <model> in this and the earlier example. Although the element names are identical, they may have different meanings. In the earlier example, model referred to the manufacturer and model together. In the latter example, model was used only for the specific model, e.g., mustang. Although the rules, or schema, for a specific XML document specify the syntax of the document, they can't ensure the 'semantic' interpretation of a document, i.e., exactly what the elements mean. Elements should be unambiguously defined in a data dictionary or similar document; alternatively, an information model such as the HL7 information model will convey explicit semantics to element names when XML documents are created based on schemas created by HL7 methodology.

[53] A default value would be assigned in the schema, or rules, for that particular type of XML document.

In this example, the element *person* has an attribute *age* with a value of 18.[54] In SPL, an example of an element containing an attribute could be:

<paragraph>This drug is <content styleCode="bold">contraindicated</content> with food.</paragraph>

The element *content* has the attribute: *emphasis*, with a value of "bold". When displayed, this code would likely be displayed as follows:

This drug is **contraindicated** with food[55]

Empty elements may be used to hold attributes and values, e.g.,

<renderMultiMedia referencedObject="MM1"/>

In this example there is no content for the *renderMultiMedia* element since it is an empty element.[56]

7.5.4 The Structure of an XML Document

XML documents generally consist of two parts, the 'processing instructions' which are at the start of an XML document, and the content or body of the document. This is not to be confused with the header and body of an SPL document which *together* form the body of an XML document.

Within the body of an XML document lays the root or parent element and all its “children” or sub-elements. This document hierarchy is often depicted using the metaphor of a tree. As described previously, elements can contain content, other elements (sub-elements or children) or both (mixed content). In XML, it is imperative that the children or sub-elements be nested properly.[57] Using the previous example:

<paragraph>this drug is <content styleCode="bold">contraindicated </content> with food.</paragraph>

The *paragraph* element is mixed, because it contains text content as well as other elements. The element *content* is nested within the paragraph.

The element *text* in the SPL example below is an illustration of an element that contains only other elements[58]:

<text>
<paragraph>This is just the paragraph content</paragraph>
<renderMultiMedia referencedObject="MM5"/>
</text>>

[54] The reader may ask why this isn't an element, e.g., <person><name>Toni</name> <age>18</age></person>. How to make the decision whether a certain value should be specified as an attribute or element in the structure of an XML document is beyond the scope of this primer.

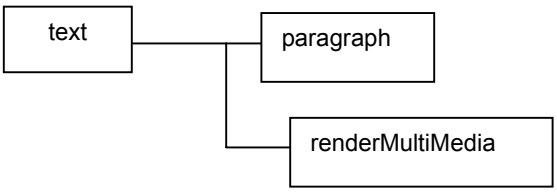
[55] The process whereby the word contraindicated is displayed as bolded text is a property of the stylesheet and rendering method that interprets 'bold' to mean 'bold the content of this element when it is displayed.' It is not a property of XML or the SPL document per se. The stylesheet could just as easily interpret 'bold' as 'important' and display it as italicized text; however, within SPL the semantic meaning of bold is to bold the element when displayed so that is what the stylesheet is designed to do.

[56] For more information regarding the <renderMultimedia> element see Section 4.1.6.

[57] Nesting in this context refers to the order of the children of an element, i.e., that an actual XML document follows the structure permitted in the rules (or schema) for that document.

[58] Text is being used indifferent contexts in this discussion. Text refers to the content, or value of an element (e.g., the text, or value, or content of the *paragraph* element in <paragraph>green olive</paragraph> is 'green olive'. *text* is also an element name in SPL as in <text><paragraph>green olive</paragraph></text>.

In this example, the element *text* contains the elements *paragraph* and *renderMultimedia*. p*aragraph* and *renderMultimedia* are children or sub-elements of *text*. Using the tree metaphor to represent this relationship, we have:



7.5.5 XML Instructions and the Root Element

XML documents contain instructions that do not contain either content or data elements, but are used by the program processing the XML document for different purposes (e.g., a web browser opening an XML document). The XML declaration at the start of an XML document identifies the document as an XML document and specifies the version of the XML standard used when creating the document. Other processing instructions may be included that, for example, specify a transformation file that will aid in displaying the XML content. The following is an example of a declaration plus processing instructions in an XML document that specifies a transformation file:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="spl-1.0.xsl "?>
```

The ? after the left angle bracket identifies each line of this markup as an XML instruction.

The root element of an XML document is a required and critical aspect of an XML document as it contains all the other elements that constitute the document (i.e., it defines the starting point and ending point for the document). The root element uses a special syntax and identifies the schema(s) that will be used to validate an XML document.

The root element has many optional attributes, the description of which is beyond the scope of this primer, and may often appear quite complex. The specific form of the root element to be used with SPL is specified in the implementation guide above. However, the general order of XML document is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ? Other Processing instructions… "? -->
<rootElement>       <!-- start tag for root element of document -->

….
….
</rootElement>      <!-- end tag for root element -->
```

7.5.6 XML Comments

Comments may exist almost anywhere in an XML document except within tags, i.e., inside markup. Comments are ignored by the XML processor. Comments are identified by the special character sequence <!- - (start comment) and --> (end comment). For example:

```
<example><!-- this is a comment --></example>
```

A more detailed example of valid XML comments is:

```
<text>
        <!--  this was written to address Matt's concerns -->
        <paragraph>drug x may cause…. </paragraph>
```

```
        <paragraph><! -- another comment, but this time mixed in the content of an element -->drug
        x may also cause…. </paragraph>
</text>
```

The following would not be permissible because the comment is inside the tag:

```
<text>
        <paragraph <!-- this is a comment --> styleCode="bold">this is invalid since the comment is in
        markup (in this case the start tag for paragraph)……</paragraph>
</text>
```

7.5.7 XML Schemas

A schema is a separate XML document that describes the rules, or syntax, of an XML document type (e.g., SPL). A schema defines the permissible elements and attributes in a document, the order in which they occur in an XML document, and possible values for these elements[59]. For example, a schema might permit:

```
<text><paragraph>this is OK</paragraph></text>
```

but not permit:

```
<paragraph><text>this is not OK</text></paragraph>
```

or just as easily, the schema could permit the reverse.

The schema also specifies the values that can be used for attributes, where the value is restricted, and can even specify the type of content that is acceptable for the value (the data type), e.g., specifying certain attributes to be dates and others only as numbers.

XML schemas can be straightforward or extremely complex depending on the document type the schema is intended to address. Creating schemas is beyond the scope of this primer. However, it should be noted that authors of XML documents (e.g., SPL authors) would not alter a schema in creating a document; changes to a schema can only be made by the creator of the schema. Authors, however, may frequently refer to the schema to understand the structure of a document and determine what elements and attributes are permissible for that document type.

7.5.8 Well formed and Valid XML Documents

An XML document must be well-formed. A well-formed XML document follows the basic rules of every XML document; the rules are the XML standard and are independent of the schema used for a particular document type. These rules apply to all XML documents regardless of the schema, i.e., that all elements have a start tag and end tag (except 'empty tags'), that nesting is correct, that attribute values are in quotes, etc. For example:

```
<test><paragraph>paragraph content…</paragraph></test>
```

is well formed.

```
<test><paragraph>paragraph content…</test>
```

[59] Another means of specifying the rules, or the syntax of XML document is a DTD, or document type definition. Both are similar in that they define the rules of an XML document, (a DTD is actually a type of schema) but the details of each differ substantially. Discussion of the details of both schemas and DTDs is beyond the scope of this primer, although it is important to note that schemas rather that DTDs are used to define the structure of an SPL document.

is not well-formed since there is a missing end tag for the paragraph element.

Similarly,

```
<test><paragraph>paragraph content….</test></paragraph>
```

is not well-formed since the tags are improperly nested.

Only if a document is well-formed it can be tested for validity. Validity tests whether an XML document follows the rules of the specific schema for that document type; if it does, then, it is considered a valid document. For example, in SPL:

```
<text><paragraph>this is a paragraph</paragraph></text>
```

is valid but:

```
<paragraph><text>this is an invalid paragraph</text></paragraph>
```

is invalid since the order of elements in the latter example is not permitted by the SPL schema. However, both examples are well-formed XML.[60]

7.5.9 Tables

Tables are not part of the XML standard; they are defined by the schema of a document type that includes tables. Tables can take different forms in different document types, i.e., the permitted children of the *table* element can de different in different schema. Because tables are so important in SPL, the structure of the *table* element in SPL is discussed briefly below.[61]

The table element in SPL is identified by the <table> tag. A table may contain rows and columns. The number of columns in a table is defined by the number of cells in a row. '*Tr'* identifies row elements and the cells in a row are *td* elements. An empty cell would be <td/>.

A simple table (including comments) could be:

```
<table>
        <tbody><!-- this defines the section as the body of a table rather than the header or footer
sections  -->
        <tr>  <!-- row 1  -->
                <td>col1 value for this row</td><td>col2 value for this row</td>
        </tr> <!-- end tag for row 1 -->
        <tr>  <!-- row 2  -->
                <td>col1 value for this row</td><td>col2 value for this row</td>
        </tr> <!-- end tag for row 2 -->
        </tbody>
</table>
```

A heading row in a table is the *th* element. Use is identical to the *td cell* element, e.g.,

```
 <table>
        <thead>
```

[60] There is no requirement that XML documents be created according to a schema, i.e., a well-formed XML document is an XML document even if not created according to a schema. However, without a schema, it would be difficult to insure that any two people creating similar documents did so according to the same rules (i.e., it would be difficult, if not impossible, to insure the documents created without using the same schema would be 'compatible').

[61] Although the *table* element in SPL is very similar to the *table* element in HTML, it is not identical. The SPL schema must be used for constructing tables in SPL.

```
                <tr>  <!-- a heading row (there can be more than 1  -->
                    <th>column 1 heading</th><th>column 2 heading</th>
                </tr> <!-- end tag for heading row 1 -->
            </thead>
            <tbody>
                <tr>  <!-- row 1  -->
                    <td>col1 value for row</td><td>col2 value for this row</td>
                </tr> <!-- end tag for row 1 -->
                <tr>  <!-- row 2  -->
                    <td>col1 value for row</td><td>col2 value for this row</td>
                </tr> <!-- end tag for row 2 -->
            </tbody>
    </table>
```

When headings are used, it is regarded as good practice to divide the table into heading and body sections by use of the '*thead'* and '*tbody'* elements, as shown in the example above. (In SPL, the <thead> element is optional but <tbody> is required.)

If footnotes are to be included, they must be addressed as a separate section at the start of the table, even though they are displayed at the end of the table. For example:

```
<thead>
…..
</thead>
<tfoot>
    <tr>
        <td align="left" colspan="5">
            <paragraph> <caption>a</caption> Half-life for patients receiving a short infusion (&lt;70
            min).
            </paragraph>
        </td>
    </tr>
</tfoot>
<tbody>
…..
</tbody>
```

Presented below is an example that illustrates several aspects of an SPL table model that includes the use of table footnotes (*tfoot* element): (1) Cells in a row, header, or footnote can contain an align attribute, (2) cells can contain the colspan attribute, and (3) cells can contain *content, linkHtml*, or other constructs (see Section 7.11, Technical Note: CDA (SPL) Narrative Block DTD). Colspan allows a cell to span multiple columns. In a five column table, 5 'footnote' columns would make little sense; colspan allows the footnote to appear as anticipated, i.e., across all columns of a table in a separate row. Colspan is also frequently used in the body of a table where a particular row may need to combine columns for presentation. For example:

Column header for 1 and 2		Column head 3	Column head 4	Column head 5
Male[a]	**Female**			
4	5	100	300	500
6	7	388	887	543
P = .007		P=.34		
[a] This is an example of a footnote				

```
<table>
  <thead>
      <tr>
          <th colspan="2">Column header for 1 and 2</th>
```

```
                <th>Column head 2</th>
                <th>Column head 3</th>
                <th>Column head 4</th>
                <th>Column head 5</th>
            </tr>
            <tr>
                <th align="center" styleCode=”italics”>Male<sup>a</sup></th>
                <th align="center"><content styleCode=”bold”>female</content></th>
                <th colspan="3"></th>
            </tr>
        </thead>
        <tfoot>
            <tr>
                <td colspan=”5”><caption>a</caption>This is an example of a footnote</td>
            </tr>
        </tfoot>
        <tbody>
            <tr>
                <td>4</td>
                <td>5</td>
                <td>100</td>
                <td>300</td>
                <td>500</td>
            </tr>
            <tr>
                <td>6</td>
                <td>7</td>
                <td>388</td>
                <td>887</td>
                <td>543</td>
            </tr>
            <tr>
                <td align="center" colpsan="2">P=.007</td>
                <td colspan="3">P=.34</td>
            </tr>
        </tbody>
    </table>
```

The table has two header rows; in the second header row, the code that yields merging, or spanning the last 3 columns is: <th align="center">Male^a</th><th align="center">female</th><th colspan="3"></th>. For the fifth row, similar code is: <tr><td align="center" colpsan="2">P=.007</td><td colspan="3">P=.34</td></tr>.

The *rowspan* attribute exists for spanning rows of a table: use is similar to colpan, e.g., <td rowspan="2'>this spans 2 rows<td>.

A table may have a caption by use of the *caption* element, e.g., <table><caption>This is the caption for this table</caption><thead>…… </table>.

Two additional elements are available for organizing the structure of SPL tables: *colgroup* , with the associated *col* element. Colgroup is used organize the columns of a table so they can be presented as a group, e.g., with a heavier vertical rule between groups. Colgroup can be used as follows:

```
<table>
        <colgroup>
                <col align="left"/>
                <col align="center"/>
        </colgroup>
```

```
        <colgroup>
            <col align="right"/>
            <col align=""left"/>
        </colgroup>
<thead>
            …….
            …….
        </thead>

        <tbody>
            …….
            …….
        </tbody>
</table>
```

This defines a table with 4 columns, 2 column groups with 2 columns each. The default alignment for each column is specified (optionally) by the align attribute. Note that the *col* element can appear without a *colgroup* element, e.g., <table><col/><col/>…..

7.6 LOINC codes for SPL

Table 17: LOINC Codes in SPL

LOINC code	SPL displayName
	Document type
34391-3	HUMAN PRESCRIPTION DRUG LABELING
	Section type
34066-1	BOXED WARNING SECTION
34067-9	INDICATIONS & USAGE SECTION
34068-7	DOSAGE & ADMINISTRATION SECTION
34069-5	HOW SUPPLIED SECTION
34070-3	CONTRAINDICATIONS SECTION
34071-1	WARNINGS SECTION
34072-9	GENERAL PRECAUTIONS SECTION
34073-7	DRUG INTERACTIONS SECTION
34074-5	DRUG &OR LABORATORY TEST INTERACTIONS SECTION
34075-2	LABORATORY TESTS SECTION
34076-0	INFORMATION FOR PATIENTS SECTION
34077-8	TERATOGENIC EFFECTS SECTION
34078-6	NONTERATOGENIC EFFECTS SECTION
34079-4	LABOR & DELIVERY SECTION
34080-2	NURSING MOTHERS SECTION
34081-0	PEDIATRIC USE SECTION
34082-8	GERIATRIC USE SECTION
34083-6	CARCINOGENESIS & MUTAGENESIS & IMPAIRMENT OF FERTILITY SECTION
34084-4	ADVERSE REACTIONS SECTION
34085-1	CONTROLLED SUBSTANCE SECTION
34086-9	ABUSE SECTION
34087-7	DEPENDENCE SECTION
34088-5	OVERDOSAGE SECTION
34089-3	DESCRIPTION SECTION
34090-1	CLINICAL PHARMACOLOGY SECTION
34091-9	ANIMAL PHARMACOLOGY &OR TOXICOLOGY SECTION
34092-7	CLINICAL STUDIES SECTION
34093-5	REFERENCES SECTION
34391-3	HUMAN PRESCRIPTION DRUG LABELING
38056-8	SUPPLEMENTAL PATIENT MATERIAL
TBD	PATIENT PACKAGE INSERT**
TBD	MEDGUIDE**

**The Patient Package Insert and MedGuide LOINC codes can only be used as subsections for the Supplemental Patient Material Section. These two codes should not be used unless nested within a Supplemental Patient Material section, e.g.,

```
<component>
    <section>
        <id root="5031A439-E76A-4FEB-827D-9AC5A758076A "/>
        <code code="38056-8" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
        displayName=" SUPPLEMENTAL PATIENT MATERIAL"/>
        <component>
            <section ID="87961815-06CE-47CA-A2D2-EFB2F24EFA44">
                <id root="51A61214-46FE-779A-F2E2-AF5F242FC44"/>
                <code code="TBD" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
                displayName=" PATIENT PACKAGE INSERT"/>
                    <text>
                        <paragraph>…..</paragraph>
                    </text>
            </section>
        </component>
    </section>
</component>
```

7.7 Organization of files for submission to FDA

SPL submitted to FDA should have all files organized at one level, i.e., there should not be relative or nested sudirectories. All files should be included in a folder named 'SPL'. The following principles should be followed:

1. The name of the xml file containing the spl file must have the *.xml extension. It is preferable the file name reflect the proprietary name of the product or a variation, e.g., "toprol-xl.xml".
2. Different versions of the SPL files should be indicated by version information inside the file but not in the file name. The *.xml file name should be identical in all submissions/revisions of the document sent to FDA.
3. Each submission of the *.xml file to FDA should include all supporting files, i.e., each submission should include all associated image files even if these have not changed.
4. The *.xsl file used to view the rendered *.xml file should be included with the submitted files.[62]
5. Multi-word file names are built using dashes, e.g., "toprol-xl.xml"
6. Absolute URLs or URLs containing scheme identifiers (file: or http:) or any path names (e.g., "..", or "graphics/") should not be used in SPL.

An example set of files submitted to FDA might consist of the following:

toprol-xl.xml
toprol-figure-01.jpg
toprol-figure-02.jpg
toprol-figure-03.jpg
spl-2a.xsl[63]

[62] Note: this is only required prior to development of ELIPS. At that time it is anticipated that the requirement for inclusion of the *.xsl file with the submission will be eliminated.
[63] See previous footnote.

7.8 Identifiers for FDA Data Elements

Earlier, pre-release versions of the SPL implementation guide identified four separate FDA-maintained tables to be used as lookup tables for certain data element values in Section 5.2. These included:

(a) Dosage Form (form code): This standard provides for all drug dosage forms. CDER Data Element Number: C-DRG-00201, http://www.fda.gov/cder/dsm/drg/drg00201.htm.

Example usage:

```
<formCode  code="value from DRG 201" codeSystemName="FDA OID for Table DRG 201"
codeSystemName="Name for Table DRG 201" displayName="text for value from DRG 201"/>
```

(b) Labeled Route of Administration (route code): This standard provides for all routes of administration for drugs. CDER Data Element Number: C-DRG-00301, http://www.fda.gov/cder/dsm/drg/drg00301.htm

Example Usage:

```
<routeCode code="value from table DRG 301"  codeSystem="OID for DRG 301" codeSystemName="Name
for table DRG 301" displayName="text for value from DRG 301">
```

(c) Package Type (form code): This standard provides for all types of packages that bulk drug substances and final drug dosage forms are contained in, including both the immediate (or primary) and secondary containers, CDER Data Element Number: C-DRG-00907, http://www.fda.gov/cder/dsm/drg/drg00907.htm

Example usage:

```
 <containingPackagedMedicine>
        <formCode code="value from table DRG 907"  codeSystem="OID for FDA Table 907"
        codeSystemName="Name for table DRG 907" displayName="text for value from DRG 907">
 <code code="0002-7502-01" codeSystem="2.16.840.1.113883.6.69" codeSystemName="NDC"/>
 </containingPackagedMedicine>
```

(d) Ingredient Units: This standard provides for all drug potencies. CDER Data Element Number. C-DRG-00501, http://www.fda.gov/cder/dsm/DRG/drg00501.htm.

Example usage:

```
<activeIngredient>
        <quantity>
          <numerator value="200" unit="MG"/>
          <denominator value="1"/>
      </quantity>
<activeIngredient>
```

It is anticipated that towards the end of Q2 2005, information from the tables listed above (i.e., C-DRG-00201, C-DRG-00301, C-DRG-00501, and C-DRG-00907) will be fully integrated into the National Cancer Institute Thesaurus, maintained by the NCI Center for Bioinformatics. Information regarding the NCI Thesaurus is available at http://nciterms.nci.nih.gov/NCIBrowser/Startup.do. Change control for the 'tables' that contain values appropriate for these data elements will be managed jointly by FDA and NCI.

Although the FDA tables will become integrated within the NCI, these tables will continue to be retrievable as an FDA 'tagged-table' property within the thesaurus; concept codes for each element will identify the 'virtual tables' within the thesaurus. Unique codes will identify each possible value for each data element; these values are unique across the NCI thesaurus.

Accordingly, after announcement of the integration of the FDA tables in the NCI thesaurus, the OID value for the NCI Thesaurus (2.16.840.1.113883.3.26.1.1) should be used for the codeSystem value for each of the data elements listed above and the value for 'code' should the appropriate unique value from the NCI thesaurus.

For example, for Route of Administration:

<routeCode code="code value of identifier from NCI thesaurus (from subset of values with Route of Administration concept code)" codeSystem="OID for NCI thesaurus, i.e., 2.16.840.1.113883.3.26.1.1" codeSystemName="code system name for NCI thesaurus" displayName="name of identifier from NCI thesaurus">[64]

An example from an unrelated domain in the NCI thesaurus is the concept of 'Phlebotomy' with the identifier "Phlebotomy" and the value 'C28221'.

The NCI thesaurus is available in either downloadable form or through various APIs when integrated in an application. Additional information is available at http://ncicb.nci.nih.gov/core/caBIO. Additional information regarding the interface for retrieving FDA-specific tables for SPL from the NCI thesaurus will be posted to http://www.fda.gov/oc/datacouncil/spl.html when available.

[64] As discussed previously, codeSystemName and displayName attributes are redundant information and are generally unnecessary with submission.

7.9 Technical Note: The Nature and Use of Identifiers in SPL

The term “identifier”, or “id” for short, has several definitions in the context of SPL. It may refer to:

(a) the <id> element defined in the schema;
(b) the <setid> element defined in the schema;
(c) the XML *id* attribute defined for certain elements, e.g., <section> (written as “<… *id*>”); or,
(d) an object identifier, referred to as an OID or UUID, used as a value of specified attributes in certain elements.

Throughout this section the generic term “id” will not be used; the specific element or attribute names, or the terms OID or UUID will be used to make the meaning clear.

7.9.1 The <id> element

Purpose: The <id> element is used to uniquely identify a single logical instance of an element in (or of) an SPL instance, by means of the value of the <id *root*> attribute.

The value of <id *root*> is defined to be a Unique Identifier (i.e., of data type *uid*; see Section 7.9.6 below) and must be *globally unique*.

7.9.1.1 <id *extension*> attribute not used

Note that the schema defines an additional, optional attribute, <id *extension*>, but that attribute is *not* to be used in SPL Instances.

7.9.2 Declarative usage of the <id> element*:*

The primary usage of the <id> element is *declarative*; that is, where it carries a value uniquely identifying the instance of the element that encloses it. This usage is found in the following constructs where an <id> element is enclosed within:

(a) the <document> element [required],
(b) a <section> element [required],
(c) an <observation> element [optional], and
(d) an <observationMedia> element [required].

7.9.2.1 <id root> attribute required

While the attribute <id *root*> is optional under the schema, the rule for the use of <id> elements in SPL is that wherever an <id> element is used it *must* carry a globally unique identifier in the <id *root*> value.

7.9.2.2 bit image identification

Note that the globally unique identifier identifies a single logical instance of the element in question, i.e., a specific *bit image* of the element, not a specific copy. Any change, however trivial, to the content of that element creates a new instance of the element which must carry a different globally unique identifier when released by the originator of the instance.

7.9.2.3 identification only

Note also that the information carried by an <id *root*> value serves on only to distinguish that instance of the SPL element carrying that <id> from every other instance of an SPL element carrying an <id>; no other meaning can be inferred from the value, and no other use can be made of it.

7.9.3 Referential usage of the <id> element:

A secondary usage of the <id> element is referential, i.e., where it carries a value identifying an element in another (i.e. external) SPL instance. This usage is found in the following constructs where the <id> element is enclosed within:

(a) a <relatedDocument> element.
(b) a <sectionReplaced> element.

The <id *root*> value is that of a <document><id> or <section><id> in another (i.e. external) instance.

7.9.4 The <setid> element

The <setid> element is used to uniquely identify a set of sequentially-related SPL document instances. It carries an attribute, <setid *root*>, which functions here as in an <id> element, and carries the value of the <document><id> of the first document in the set.

7.9.4.1 Unique identifier required:

While the attribute <setid *root*> is optional under the schema, the rule for the use of the <setid> element in SPL is that wherever it is used it *must* carry a globally unique identifier in the <setid *root*> value.

7.9.4.2 <setid root EXTENSION> attribute not used

Note that the schema defines an additional, optional attribute <setid *extension*>, but that attribute is ***not*** to be used in SPL Instances. It will be deleted from later versions of the SPL Specification and Schema.

7.9.4.3 Referential Usage Only

The <setid> element is always used referentially. The value in the <setid *root* > attribute must always refer to the values in an external <document><id> element; therefore the first document in a set must not carry a self-referencing <setid>.

Note that the SPL specification states that a <setid> element may be present in the first occurrence of a set, but this practice is *not* to be followed in SPL instances. The statement will be deleted from the next version of the Specification.

7.9.5 The XML <… id> attribute

A number of SPL elements may have an <… *id*> attribute (or an attribute with another name of type *xs:ID*) that may be given a value to uniquely identify the particular element instance within the host SPL instance. The element bearing the <… *id*> may be referenced by means of the same value in an <… *idref*> attribute (or an attribute with another name of type *xs:IDREFS*) in another element within the same instance.

The important use of *id/idref* style relationships in an SPL instance is to relate references to the rendering of graphics in the narrative text to the definition of the source file for the corresponding graphic held in a separate element. The <observationMedia *ID*> attribute is of type *xs:ID* and its value identifies the element, which carries the definition of a graphic file; the<renderMultiMedia *referencedObject*> attribute is of type xs:IDREFS and its value identifies the value of the <observationMedia *ID*> attribute in the appropriate <observationMedia> element.

7.9.5.1 no cross reference to <content> elements

The SPL standard also discusses a possible use for <content i*d*> and <reference *value*> relationships to "import" the content of a given <content> element appearing in the narrative text into an element in the

structured data. This discussion should be regarded only as an illustration; the use of <content *id*> and <reference *value*> relationships to represent narrative text inside structured data is **not** permitted in SPL instances at the present time.

7.9.6 Unique Identifiers

Unique Identifiers are of HL7 data type *uid*, and can be instantiated by values derived from one of two different schemes for creating unique identifiers, either:

(a) Universally Unique Identifiers (UUID's) also known as Globally Unique Identifiers (GUID's), created according to an IETF scheme; or,
(b) Object Identifiers (OID's), created according to the ISO/ITU Object Identification scheme.

Unique identifiers are to be used in the following constructs in SPL:

(c) as values of <… *root*> attributes, and
(d) as values of <… *codeSystem*> attributes.

In declarative <id> elements it is strongly *recommended* that a UUID/GUID be used as the value of the <id root> attribute, but OID's *may* be used.

In all other cases the unique identifiers are used referentially, and always take the value used externally to declare the unique identity of the thing referenced, whether that value is an OID or a UUID/GUID.

7.9.6.1 UUID/GUID's

Universally /Globally Unique Identifiers are generated algorithmically on a computer according to the specification set out in the Internet Engineering Task Force (IETF) Working Draft, "UUIDs and GUIDs", which can be found at http://www.ics.uci.edu/~ejw/authoring/uuid-guid/draft-leach-uuids-guids-01.txt

All computer operating environments have available utilities that implement the algorithm to generate UUID/GUID's with the property that each generated instance is reliably unique among all other UUID/GUID's generated by the same algorithm, regardless of where and when it is created.

The UUID/GUID is a 128-bit binary value, represented in an attribute value as a hexadecimal string (i.e. UTF-8 encodings of the digits 0-9, and letters A-F, each representing 4-bit segments of the value)

7.9.6.2 OID's

The ISO/ITU-T Object Identifier (OID) scheme is a dotted notation (d.d.d …), where each "d" is a positive integer of one or more digits, representing a tree structure; that is, an OID represents a node in a tree in terms of numbered branches starting from the root of that tree.

By convention, *nationally registered* organization ID's will be used in SPL as the initial (left most) portion of the root attribute value to distinguish the numbering domains of each organization

For corporations registered in the US the left most portion of the root attribute will be "2.16.840.1.nnnnnn" where

2	= joint ISO/ITU-T assigned codes
16	= country assignments
840	= United States
1	= United States Companies/Organizations
nnnnnn	= Company ID

(the examples shown in the SPL specification use "2.16.1.840.1.113883", where 113883 is HL7)

Organizations may apply for a Company ID number from the *Organizational Name Registry* operated by the American National Standards Institute (ANSI). There is a fee for this service. Details can be found at:
http://www.ansi.org/other_services/registration_programs/reg_org.aspx?menuid=10

Organizations may also apply for a Company ID number from HL7, which has authority from ANSI to distribute such ID's within its Company root. These are of the form
"2.16.840.1. 113883.3.nnnn" where

2	= joint ISO/ITU-T assigned codes
16	= country assignments
840	= United Stated
1	= US Companies
113883	= HL7
3	= HL7 Company/Organizational Registry
nnnn	= Company ID

Details can be found at http://www.hl7.org/

7.9.6.3 Declarative Use of Unique Identifiers (in <id ROOT> attributes)

Unique identifiers are used declaratively only in <id *root*> attribute values. The creators of SPL instances are responsible for ensuring that each <id *root*> value used in an SPL instance is globally unique. This implies that the values will be created and managed by some automated process; creation by human users would simply be too prone to error.

See Section 7.9.7, Document and Section Identification, for details of the use of <id> elements in SPL life cycle management.

7.9.6.3.1 Declarative Use of UUID/GUID's (in <ID *ROOT*> attributes)

UUID/GUID's are recommended for declarative use as <id *root*> attribute values because they are easily generated using widely-available software and each instance is reliably unique among all other UUID/GUID's generated by the standard algorithm at creation time.

7.9.6.3.2 Declarative Use of OID's (in < ID *ROOT* > attributes)

If an SPL author chooses to use OID's for declarative use as <id *root*> attribute values, the author will have to obtain a registered OID for their own use as the root of a tree structure on which they construct unique identifiers. This implies the implementation of some mechanism to manage the dotted values to the right of their registered OID root, and ensure that each instance used is unique; that is, some common software and database service must be available to all programs that use an organization's OID's as unique identifiers to ensure that each one assigned is allocated only once and never reused.

This mechanism, for example, might create a sub-tree following its company number with a numbered branch for each subsidiary company, and/or a sub-tree with a numbered branch for each drug product. Such a mechanism might be useful as part of a broader corporate document management scheme, but it is in no way required. An organization is only obliged to ensure that the entire OID string in the <id *root*> attribute value is unique for each declarative <id> element instance that it issues. However, since this ID mechanism may also be used on other documents or software objects, it is important that each organization ensure that whatever scheme they use does not allow for duplicating OID's even on different classes of documents or objects: these ids must be *globally* unique!

There is no context in SPL in which the use of an OID in a declarative <id> element carries any reliable information about the organization legally responsible for the issuance of an SPL instance. For example, an OID used in the <author><AssignedEnity><id> element in the SPL document header might be misconstrued to identify the entity in question; however, a company headquartered in

Switzerland might obtain its Company ID there (2.16.756. …) and allocate a sub-tree branch number to a US subsidiary. Or, a company providing regulatory affairs services might create documents on behalf of another company, using its own Company ID and its own OID coding scheme.

From the foregoing it should be obvious that the use of OID's in declarative <id> elements is more complicated than the use of UUID/GUID's, though there may be other values in their use to offset the complexity in some application contexts. At this time, FDA is recommending that only UUID/GUID's be used in <id> elements SPL instances.

7.9.6.3.3 Use of Unique Identifiers in <… *codeSystem*> attributes

In an SPL instance external, predefined code systems must be used in association with the following elements:

(a) <code>
(b) <formCode>
(c) <routeCode>
(d) <languageCode>
(e) <confidentialityCode>
(f) <signatureCode>

Each of these elements has a <… *codeSystem*> attribute, which by HL7 definition takes an OID as a value. The particular code systems to be used in SPL are specified by the regulatory. The OID's that identify these are externally predefined, like the code system they reference; an alternative way to describe this is that each code system is identified by an externally registered OID. See section 7.10, Technical Note: The Nature and Use of Code Systems in SPL for more details on codes and code systems.

Since code systems are always externally defined, the use of <… *codeSystem*> attribute values is always referential.

By convention, code systems used under HL7 standards use an OID as the registered identifier of a code system.

7.9.7 Document and Section Identification

Both the <document> element and the <section> element each require a globally-unique instance identifier, where "instance" means that the entire digital content (as a bit string) of the enclosing element from the opening "<" of the start tag to the closing ">" of the end tag is unique; i.e. it constitutes a single, specific version of the content. This globally unique identification is expressed as the value of an <id *root*> attribute and should be a UUID/GUID as described in section 7.8.6.1.

Good practice suggests that instance identifiers be managed and assigned by automated processes integrated into organization's document management and/or document authoring software.

7.9.7.1 Identification Within Structured Data

Within structured data each <section> element must have an <id> element. These <id> elements have the same purpose and are subject to the same rules as noted above for <section>elements in narrative text. This applies to the <id> elements in the SPL Structured Data Section, each SPL Drug Product Description Subsection, and each Drug Product Component Description Subsection.

The schema also provides for an optional <manufacturedProduct><id> element, but it is *not* to be used in SPL instances.

7.9.8 Document Versioning

The SPL standard defines the <versionNumber> and <setid> elements as optional parts of the document header. Version numbering is a well-established and well-understood practice. Since the document ID needs to be unique but does not necessarily contain information that could be used to sequence successive versions (and might be a relatively long number in any case), good practice suggests that a user-friendly version number be used.

Each different instance of a document released by an originator must carry a value in the <versionNumber> element that is numerically greater than the version number of any previously released instance. As seen by any receiver the sequence does *not* need to be dense (i.e. numbers may be missing, for example, numbers used on drafts made internally that were never released), but it does need to be orderable in the temporal sequence of release.

7.9.9 Summary of Identification Markup for Updates of Whole SPL Instances

FDA will be publishing a separate reference document addressing versioning and life cycle requirements for SPL. It is anticipated this will be available in early May at http://www.fda.gov/oc/datacouncil/spl.html.

Each time a new version of an SPL instance is released it must carry:

(a) a new <section><id *root*> attribute value for each changed section. The schema provides for a <section> <sectionReplaced> element carrying the <id> of the section replaced in the previously released version but this element is not used by FDA at the present time..
(b) a new <section><id *root*> attribute value for each new (inserted) section.
(c) a new <section><id *root*> attribute value for each section, if any, enclosing a changed or inserted section..
(d) a <section><replacementOf><sectionReplaced><id *root*> carrying the <id> of the enclosing section replaced in the previously released version.. This element is not used by FDA at the present time.
(e) a new <document><id *root*> attribute value for the whole instance..
(f) a <setid> element with the value being the <document> <id> of the first released instance.
(g) If a section is unchanged from a previous submission in a new submission then the identifier/GUID should be unchanged as well.

7.10 Technical Note: The Nature and Use of Code Systems in SPL

Code systems are used as attribute values in the following elements:

(a) <code>
(b) <formCode>
(c) <routeCode>
(d) <languageCode>
(e) <confidentialityCode>
(f) <signatureCode>

(Note that code systems used in the definition of abstract data types used in the schema, notably class codes and mood codes defined for XML attribute values, are not used explicitly in an SPL instance and are therefore not discussed in the Implementation Guide.)

In addition, the <code> element itself is conditioned by its enclosing element, and is defined in SPL in the following contexts:

(a) <document><code>
(b) <section><code>
(c) <manufacturedMedicine><code>
(d) <monitoringProgramEvent><code>
(e) <policy><code>
(f) <activeMoiety><code>
(g) <genericMedicine><code>
(h) <activeIngredientSubstance><code>
(i) <containerPackagedMedicine><code>
(j) <observation><code>

7.10.1 Required Attributes

All elements that use code systems are defined as carrying the following attributes:

(a) code (code value from the code system applicable to the specific occurrence)
(b) codeSystem (OID, see section 7.8.6.2)
(c) codeSystemName (string, formal name of system)
(d) codeSystemVersion (string, version used)
(e) displayName (string, text associated with code value, from code system)

While all of these attributes are optional under the schema, the rule for the use of the <code> element in SPL is that wherever it is used the following attributes *must* be populated correctly:

(a) code (code value from the code system applicable to the specific occurrence)
(b) codeSystem (OID, see section 7.9.6.2)
(c) codeSystemName (string, formal name of system)
(d) displayName (string, text associated with code value, from code system)

The displayName and codeSystemName can be omitted as these are determined by (and therefore redundant) to the code and codeSystem attributes. codeSystemVersion attributes should be omitted only if the code system used has no defined version number.

If the code system supplies no display name for the particular code value, the attribute should carry a null string (“”) as its value.

7.10.2 Restricted Content

In the SPL schema all elements that use a code system are defined to be of HL7 data type CE (coded with equivalences), except for <policy><code> which is defined as data type CD (Concept Descriptor). CD elements *may* contain any or all of the following optional elements: <originalText>, <translation>, <qualifier>; CE elements are derived from CD with the restriction that the <qualifier> element is not allowed.

In SPL the use of these embedded elements within a <code> element is further restricted:

(a) The <translation> element may be used within the <medicine><code> element may be used, if needed to carry additional drug codes (such as EMEA MAN's);
(b) The <text> element may be used in <characteristic><code> elements used to convey drug identification and markings, where there is no code system to express the identification adequately; however, at the present time the following construct is recommended <characteristic classCode="OBS"><id root="…"/><code code="FDA code name, e.g., FDACOLOR"/><text>e.g., white</text></characteristic>. (See Section 5.2: Coding the Data Elements.)
(c) In all other cases where elements may be contained in elements of type CE or CD according to the schema, such elements are *not* be used in an SPL instance.

7.10.3 Source of Code Systems

The code systems to be used are determined by the regulatory realm. Examples of code system used in SPL are given in the following subsection.

7.10.4 LOINC Codes

The Logical Observation Identifier Names and Codes (LOINC) is a code system maintained by the Regenstrief Institute, and used by HL7. For more details see: http://www.loinc.org/

The OID for the LOINC code system is "2.16.840.1.113883.6.1".

LOINC will be used a source of some codes to be used for SPL instances bound to US (FDA) rules. At present there are codes defined in LOINC for FDA labeling documents and for sections of US prescription labeling.

For example, the <code> element in the context <document><code> would be written like this:

```
<code value=”34391-3 “
codeSystem="2.16.840.1.113883.6.1“
codeSystemName=”LOINC”
codeSystemVersion=”2.1
displayName=”HUMAN PRESCRIPTION DRUG LABEL” />
```

For example, the <code> element in the context <section><code> for the Clinical Pharmacology section would be written like this:

```
<code value=”34090-1“
codeSystem="2.16.840.1.113883.6.1“
codeSystemName=”LOINC”
codeSystemVersion=”2.1”
displayName=”CLINICAL PHARMACOLOGY SECTION” />
```

The full set of LOINC codes currently applicable to SPL are listed in section 7.7.

(Note: it has been discussed in other contexts that the displayName is unnecessary as the information is determined by the codeSystem and codeSystemName pair; in this context it has been recommended that displayName *not* be included in the code element. However, during the test phase of SPL it is recommended this be included for additional error checking.)

7.10.5 Registration of External Vocabulary Domains with HL7

For the test phase any external vocabulary domains needed that are not currently registered with HL7 will be registered by the FDA. Implementers will not need to deal with this issue.

7.10.6 The Role of Regulatory Rules & Guidance

Each regulatory realm sets rules for how the intellectual content of labeling is to be expressed and organized. These rules act on the content of the narrative text and are largely independent of the SPL Specification and Implementation Guide, except insofar as they the order and nesting of sections.

However, each regulatory agency also specifies rules for the code systems (and their associated vocabulary of code values) that are to be used in an SPL instance for use in that regulatory realm, as follows:

(a) the code system to identify an SPL instance as an SPL document; expressed by the <*code codeSystem*> attribute value in the <document.><code> construct.

(b) the code system to identify section topics; expressed by the <code *codeSystem*> attribute value in the <section><code> construct used in narrative text.

Note that regulatory realm-specific rules for the ordering and/or nesting of sections are not part of the SPL specification, so there is no provision in the SPL specification or in the SPL implementation guide to express or to enforce such rules. The standard rendering follows the order of sections as they appear in an instance and displays the contents of title elements within sections, if present, regardless of the codes that may or may not be assigned.

(c) the code systems to identify languages, confidentiality level and signature systems; expressed by the <code *codeSystem*> attribute value in the following elements:

<languageCode>
<confidentialityCode>
<signatureCode>

(d) the codes to be used in structured data; expressed by the <… *codeSystem*> attribute of the <code> element in the following contexts:

<manufacturedMedicine><formCode>
<policy><code>
<substance><code>
<activeMoiety><code>
<genericMedcine><code>
<containerPackagedMedicine><formCode>
<containerPackagedMedicine><code>
<characteristic><code>

and by the <… *codeSystem*> attribute value in the following elements:

<formCode>
<routeCode>

For the test phase the only regulatory realm defined is the US, and the code systems to be used are listed below in Table 1.

Table 18: Code System Used in SPL

	Markup Usage	Code System Or Code Data Source	<…CodeSystem> OID
	<document><code>	LOINC: FDA Rx Labeling	2.16.840.1.113883.6.1
	<section><code>	LOINC: FDA labeling section codes	2.16.840.1.113883.6.1
	<containingPackagedMedicine><code>	NDC code	2.16.840.1.113883.6.69
	<substance><code>	UNII dictionary under development	To be assigned
	<activeMoietyEntity><code>	UNII dictionary under development	To be assigned
	<genericMedicine><code>		
	<containingPackagedMedicine><formCode>	C-DRG-00907	To be assigned
	<characteristic><code>	See Table 16: Imprint Codes	FDA
	<formcode>	C-DRG-00201	To be assigned
	<routeCode>	C-DRG-00301	To be assigned

7.11 Technical Note: CDA (SPL) Narrative Block DTD

The following is the DTD for the narrative block (text) sections of SPL. It is identical to the current Clinical Document Architecture model.

```
<!ENTITY % textAtts "
  ID ID #IMPLIED
  lang NMTOKEN #IMPLIED
 styleCode NMTOKENS #IMPLIED">

<!ELEMENT text (#PCDATA | content | linkHtml | sub | sup | br | footnote |
footnoteRef | renderMultiMedia | paragraph | list | table)*>

<!ELEMENT content (#PCDATA | content | linkHtml | sub | sup | br | footnote |
footnoteRef | renderMultiMedia)*>
<!ATTLIST content
 %textAtts;
 revised (insert | delete) #IMPLIED>

<!ELEMENT linkHtml (#PCDATA | footnote | footnoteRef)*>
<!ATTLIST linkHtml
 name CDATA #IMPLIED
 href CDATA #IMPLIED
 rel CDATA #IMPLIED
 rev CDATA #IMPLIED
 title CDATA #IMPLIED
 %textAtts; >

<!ELEMENT sub (#PCDATA)>

<!ELEMENT sup (#PCDATA)>

<!ELEMENT br EMPTY>

<!ELEMENT footnote (#PCDATA | content | linkHtml | sub | sup | br |
renderMultiMedia | paragraph | list | table)*>
<!ATTLIST footnote
 %textAtts; >

<!ELEMENT footnoteRef EMPTY>
<!ATTLIST footnoteRef
 %textAtts;
 IDREF IDREF #REQUIRED>

<!ELEMENT renderMultiMedia (caption?)>
<!ATTLIST renderMultiMedia
 referencedObject IDREFS #REQUIRED
 %textAtts; >

<!ELEMENT paragraph (#PCDATA | caption | content | linkHtml | sub | sup | br
| footnote | footnoteRef | renderMultiMedia)*>
<!ATTLIST paragraph
 %textAtts; >

<!ELEMENT list (caption?, item+)>
<!ATTLIST list
 %textAtts;
```

```
    listType (ordered | unordered) "unordered">

<!ELEMENT item (#PCDATA | caption | content | linkHtml | sub | sup | br |
footnote | footnoteRef | renderMultiMedia | paragraph | list | table)*>
<!ATTLIST item
  %textAtts; >

<!ENTITY % cellhalign "align      (left|center|right|justify|char) #IMPLIED
   char       CDATA          #IMPLIED
   charoff    CDATA          #IMPLIED">

<!ENTITY % cellvalign "valign     (top|middle|bottom|baseline) #IMPLIED">

<!ENTITY % Tframe "(void|above|below|hsides|lhs|rhs|vsides|box|border)">

<!ENTITY % Trules "(none | groups | rows | cols | all)">

<!ENTITY % Scope "(row|col|rowgroup|colgroup)">

<!ELEMENT table (caption?, (col* | colgroup*), thead?, tfoot?, tbody+)>

<!ELEMENT caption (#PCDATA | linkHtml | sub | sup | footnote | footnoteRef)*>

<!ELEMENT thead (tr)+>

<!ELEMENT tfoot (tr)+>

<!ELEMENT tbody (tr)+>

<!ELEMENT colgroup (col)*>

<!ELEMENT col EMPTY>

<!ELEMENT tr (th | td)+>

<!ELEMENT th (#PCDATA | content | linkHtml | sub | sup | br | footnote |
footnoteRef | renderMultiMedia)*>

<!ELEMENT td (#PCDATA | content | linkHtml | sub | sup | br | footnote |
footnoteRef | renderMultiMedia | paragraph | list)*>

<!ATTLIST table
  %textAtts;
  summary CDATA #IMPLIED
  width CDATA #IMPLIED
  frame %Tframe; #IMPLIED
  rules %Trules; #IMPLIED>

<!ATTLIST caption
  %textAtts; >

<!ATTLIST colgroup
  %textAtts;
  span CDATA "1"
  width CDATA #IMPLIED
  %cellhalign;
  %cellvalign; >

<!ATTLIST col
```

```
  %textAtts;
  span CDATA "1"
  width CDATA #IMPLIED
  %cellhalign;
  %cellvalign; >

<!ATTLIST thead
  %textAtts;
  %cellhalign;
  %cellvalign; >

<!ATTLIST tfoot
  %textAtts;
  %cellhalign;
  %cellvalign; >

<!ATTLIST tbody
  %textAtts;
  %cellhalign;
  %cellvalign; >

<!ATTLIST tr
  %textAtts;
  %cellhalign;
  %cellvalign; >

<!ATTLIST th
  %textAtts;
  abbr CDATA #IMPLIED
  axis CDATA #IMPLIED
  headers IDREFS #IMPLIED
  scope %Scope; #IMPLIED
  rowspan CDATA "1"
  colspan CDATA "1"
  %cellhalign;
  %cellvalign; >

<!ATTLIST td
  %textAtts;
  abbr CDATA #IMPLIED
  axis CDATA #IMPLIED
  headers IDREFS #IMPLIED
  scope %Scope; #IMPLIED
  rowspan CDATA "1"
  colspan CDATA "1"
  %cellhalign;
  %cellvalign; >
```